7 Years QL Today! Thank you!

QDT

Progress report by Jim Hunkins

Connect Digital Cameras to QDOS/SMSQ

Simon N Goodwin explains the general technology and how to get pictures out of digital cameras into your system

... and much more: BASIC, C, Assembler ...

listings and tutorials ...

Urs König (CoWo) & David Batty of Sector Software
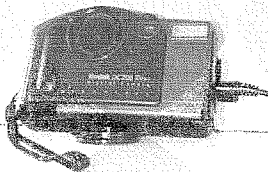
QL Personalitis we have not seen for a while at current QL shows!

Pointer Programs

A close look at some Public Domain Programs

Update on the SMSQ/E Source Code Status

Contents

Advertisers

in alphabetical order

Welcome to the seventh year of QL Today. Perhaps I should quote a Stardate after the Clocking In article in the last issue, but I'll leave matters of time to our expert contributor!

The Q60 goes from strength to strength. D&D Systems have told me that interest in the Q60 is high and they are well pleased. Who are these guys? We show you in this issue! Just before this issue went to press, Peter Graf confirmed to me that the Q40 is no more, it has been discontinued in favour of the Q40i (the improved Q40) and Q60. "The Q40 is dead, long live the Q40i."

I have had a Q60 on loan from D&D Systems for a little while now and I can tell you I am well pleased with it so far. Because we are now in the busiest of periods with a major installation on at work and working long hours as a result, I haven't managed to use it enough to warrant writing a review yet, but believe me from what I've seen so far, this is the most amazing QL system yet! The speed is breathtaking. The sheer number of things you can do with it is amazing. The 3 speaker sound system is amazing, the 20kHz sound system is worth the experience especially when you have been used to a QL BEEP! The price might seem steep at first due to the fairly low volume of manufacture, but believe me with this system you get what you pay for. Provided that the machine proves reliable long term and after sales support is adequate this machine just can't go wrong!

It may well have a Desktop or two in the near future as well. Jim Hunkins is still working very hard on QDT and has sent us a progress report, while I have been working on a less ambitious one called Launchpad which although will run on a Q60, is really designed more for QDOS users. I demonstrated it at the recent Quanta AGM and this chap came along and took an interest in it –

I didn't realise I was talking to Claus 'Q60' Graf as I had never met him before!

Quanta's recent annual general meeting revealed a few changes. John Gilpin, who has been closely associated with the North East Manchester QL group for some time and has been Quanta Head Librarian will also take on the membership secretary and treasurer roles after Bill Newell and John Taylor have stood down after many years of loyal service to the user group. And Bruce Nicholls will step into Colin Baskett's shoes as editor.

Finally, nice to see that despite being away from home for long periods with his work, Thierry Godefroy has managed to get his QDOS/SMS Repository of QL software back on line after his site vanished from the original page. Luckily, Phoebus Dokos had managed to archive a copy of the Repository and put up a temporary copy on his site. This was probably the largest single online QL software source and it's nice to see it back, Thierry!

The Mice

'Poor Eric. He got run down by a trackball on the information superhighway'
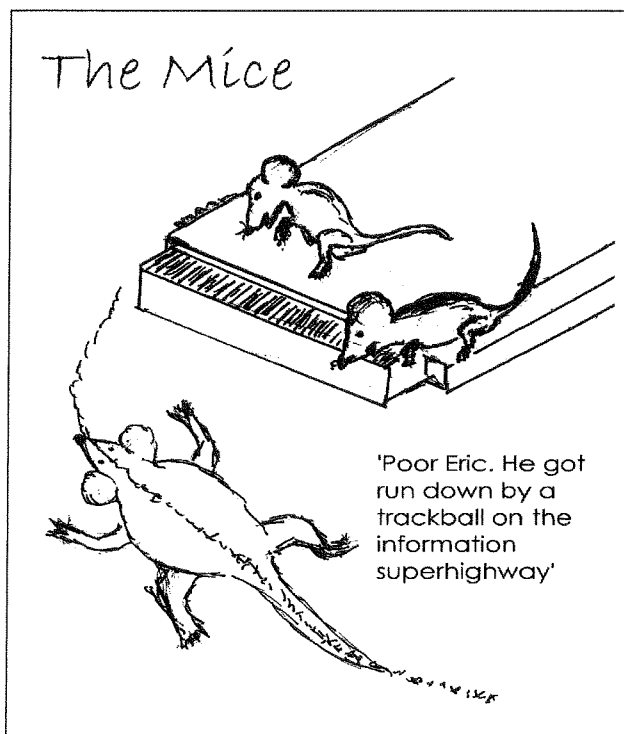
NEWS

Lear PCB-CAD V6.04

Malcolm Lear has upgraded his PCBCad program to version 6.04, which contains the following changes.

Saved states added to art file format. This extra data will simply be ignored by earlier versions. The saved states include NC drill information, thus making the separate drill file redundant. Added PCB_ prefix to all extensions. Colour schemes added, toggled by F9.

The program may be downloaded from my Other Software Page and is also available from my PD library service.

http://www.soft.net.uk/dj/software/other/other.html

PQIV Update - Claus Graf

I have finished a new version of PQIV (Q40 graphics program). The new features of v0.2 are:
- Printing to PostScript files
- Printer Dialog, size and position of image on paper can be adjusted with the mouse
- Rotation of images

Downloads at http://www.q40.de

Changes at Quanta

Bruce Nicholls will be producing the Quanta Magazine from the June Issue.

The magazine will now be electronically submitted to the printers, if you have adverts that need to be placed in the next issue please forward them on to Bruce either to his postal address given below or electronically to this email address (most QL/PC formats accepted).

Any news/reviews/articles for the next issue please also send to this email address:

editor@quanta.org.uk
Bruce Nicholls
Quanta Editor
38 Derham Gardens
Upminster
Essex
RM14 3HA
UK

At the Quanta AGM in Manchester, John Gilpin was appointed as Quanta Treasurer and Membership Secretary in addition to his existing post as Head Librarian. Email addresses are now established as follows:

quanta_librarian@uk2.net
quanta_treasurer@uk2.net
quanta_membership@uk2.net
as well as his home address: gilpins@ic24.net

The retiring officers (John Taylor and Bill Newell) have also offered to pass any messages to him during the take-over period. These changes should be made in the Quanta Magazine from the next issue.
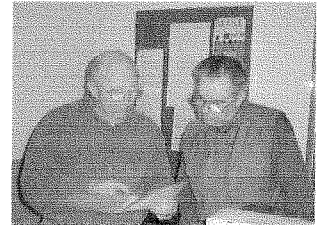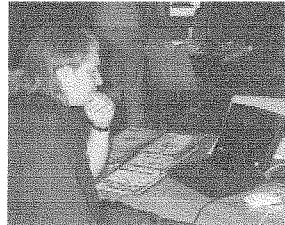
Bruce Nicholls (left picture). Former membership secretary Bill Newell (left) and John Gilpin (right picture).

QDOS/SMS Repository
Thierry Godefroy

Because of some problems with my ISP the QDOS/SMS repository has been shut down two months ago. As I could not sort out these problems, I had to setup a new web site (but I will change it again soon so as to get enough room for the whole 90Mb software archive). For the time being, the repository is available at:

http://www.multimania.com/godefroy/

Most links are in fact pointing to Phoebus R. Dokos' mirror (many thanks Phoebus for setting it up!), but the newest software I have received is now back locally as well...

As my sparse free time will permit, I will eventually get everything up and running again (an overdue update to my main QDOS/SMS site is also to come "soon")... Please be patient, everything should be sorted out by this summer...
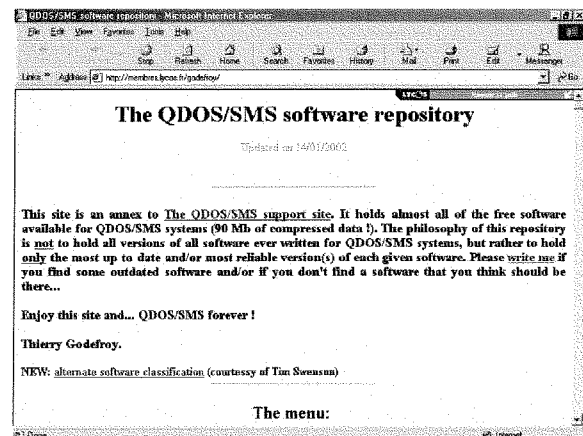
The QDOS/SMS Repository Home Page

QL/E CD

Urs Koenig has released the QL/E CD. This is a CD-ROM containing a revised QL PD/CDR distribution which includes some 600MB of software in a QXL.WIN format, pre-configured QL/E Boot Partition with the QTOP/E distribution from coWo Electronic, recent demo versions of various QL emulators and the free emulators such as QLay. The QL/E CD costs 20 Euros or 10 pounds direct from

Urs Koenig
Münsterstrasse 4
CH-6210 Sursee
Switzerland.
For further information, send an email to cowo@bluewin.ch or visit Urs's web page on http://mypage.bluewin.ch/QLvsJaguar/QL18.html

The QL/E CD

QEYBOARD

With the current scarcity of QL keyboard membranes, it is heartening to hear that Dave Park (currently based in America, known as 'Dexter' to those on the QL-Users mailing list) has been developing a new QL keyboard system called Qeyboard. He submitted a progress report to the mailing list as follows:

"I have half the PCB material, the microswitches, half the headers, ribbon cable and the diodes for the Aurora version. I also have the design completed. I'm waiting on half the headers, IDC connectors, assorted hardware (spacers, screws) and some PCB materials. I have decided that yes, I will definitely auction the first one on Ebay."

This update was dated 30th April, obviously more progress will have been made by the time you read this. With luck, Dave will have either proto-

types or completed units by the time of the USA QL show in June. No news on prices etc yet. We wish him well with this project. The current lack of availability of QL keyboard membranes should help to ensure the success of this project.

CD-ROM Readers

The release of Thierry Godefroy's Atapi/CD driver has also spawned the release of other related utilities from several authors. Intended to provide access to Atapi devices and general access to CD-ROMs on suitably equipped systems (e.g. Qubide), on some devices even audio CDs can be played. And with some of the programs on this CD, even systems not normally able to access the QXLWIN media preferred by most SMSQ systems. Any of these programs may be considered useful in their own right, together I am of the opinion that they form a really useful package.

Given the fact that several individuals on the QL scene (Phoebus Dokos, Urs Koenig, Gerard Plavec and myself for example) have released CD-ROMs for use with QL emulators such as QPC2, this means that potentially many more users can now make use of these CD-ROMs now that Qubide systems for example can access these devices. So, I have ensured that all of the following packages are available from my PD library service on 2 DD disks or 1 HD disk, library disk number GE53. The price for copying this package is £1 per disk, including supplying the media.

GE53 includes Atapi and CD-ROM system extensions by Thierry Godefroy, needed to run the other packages mentioned. QXLWIN copier for Q40/Q60 by Wolfgang Lenerz. Allows you to copy files from a QXLWIN file held on CD-ROM on a Q40/Q60 using Thierry Godefroy's CD-ROM driver extensions above. Qwirc by Per Witte. QL Winchester Information And Rename Console. QCDEZE. Duncan Neithercutt's GUI front end which enables browsing of any ISO-9660 formatted CD-ROM from a Q40 or other QL system compatible with Thierry Godefroy's ATAPI CD_ROM drivers (above) when these are installed.
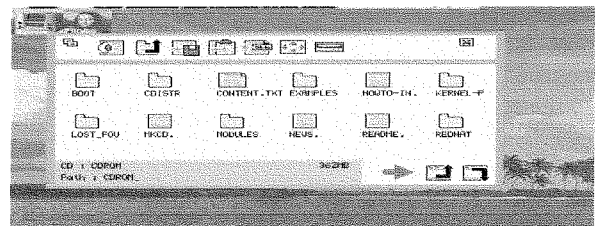
QCD_EZE pictured running on a Q40

QBOX-USA

John Impellizzeri reports that sadly he has been forced to close down QBOX-USA. The long running QL bulletin board system has become a victim of equipment failure, rising line costs and reduced useage.

John reckons that the reduced useage is probably due to more and more use being made of the world wide web and less of dedicated bulletin board systems and partly to not having enough time to allow it to be kept as up to date as it used to be. QBOX-USA was started in September 1993, and was heavily used in the mid 1990s, one of the few dedicated QL BBSes in America. John says: *"I have no regrets running it as long as we did but I think its time has come and gone."*

QL Documentation CD-ROM

Conceived the evening before the Quanta AGM in Manchester this year and actually on sale at that workshop the following day, the QL Documentation CD is a project by Darren Branagh and Dilwyn Jones to bring together as much of the publically available QL documentation as possible into one collection. Priced at just £5 (plus postage), it is also freely copyable. Documents supplied include a copy of the QL Manual in plain text format, various printer control code listing documents, hotkeys, pointer environment and operating system documentation, plus various magazine articles and many reference documents. If anyone has any such material to contribute to this work, please send files to Darren or Dilwyn, preferably by email or on floppy disk (to reduce the risk of error while retyping or scanning paper documents).
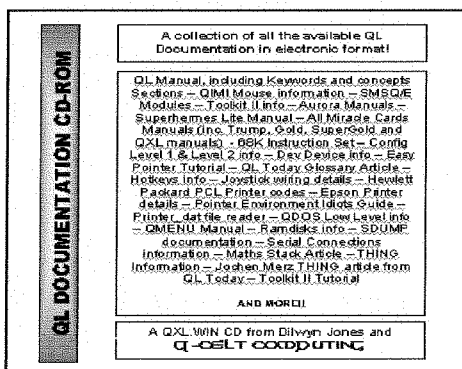

QL Documentation CD

News from George Gwilt

The programs for which I am responsible are now on the SQLUG website at
www.jms1.supanet.com
(the site is maintained by John M. Sadler)

These programs now include CPTR, which is a new system for programming PE (pointer environment) via C68. This differs from the "official" one produced by Tony Tebby.

The total of the zipped files amounts to around 1 1/2 million bytes.

Almost all the programs have been altered recently. Also the source files are now available.

Although the set of Turbo programs is my responsibility that of maintaining the manuals is Tim Swenson's and the Turbo TK code, Dave Gilham's. This means that these parts of Turbo are not on the SQLUG site.

QCOLOUR and QWIRC

Wolfgang Uhlig's QcoLour software is now available from the Other Software Page on my website. This is a programming aid to help you calculate colour numbers using on screen sliders to adjust colours. This is a program strictly for the "colour drivers" (GD2 Graphics Driver in SMSQ/E), which (to quote Roy Wood) "gives you a splash of colour" on your computer. QcoLour also introduces colour "skins" to QL programs for the first time, I believe.

Also, on the same page, Per Witte's Qwirc software is now at version 0.47.

Both programs will also be available from DJ's PD library service if you prefer to obtain the software on disk – QcoLour needs either 2xDSDD disks or 1xHD disk.

TURBO Compiler

Turbo Compiler version 4 release 15 (v4.15) from George Gwilt is now on the Other Software Page on my website.

This corrects a problem in the use of TO with CON/SCR channels.

http://www.soft.net.uk/dj/software/other/other.html
or from the SQLUG website as mentioned above.

Mark Knight Website

QL software author Mark Knight has now set up his own web site for his QL software and his work on computer graphics generally. Visit it on:
http://www.the-furnace.demon.co.uk

Q-TRANS

Q-Trans is a new pointer driven dual window file copier, by Dilwyn Jones. What makes this one different is that it has both source and destination windows, you can see the contents of both the drive being copied from and the drive being copied to. Simply select the files to be copied in the appropriate window, click on the arrow bet-

ween the two file list windows and files will be copied. The copier has the usual overwrite options, along with Delete, Execute, View and Print options like the QPAC2 files menu. Quick selection of drive name and number via icons possible, along with awareness of Toolkit 2 defaults. Its display can be resized to suit displays from QL sized right up to 768 pixels high!

Above all, it's designed to be simple to use! Q-Trans is part of a forthcoming QL Desktop project for QDOS systems, but capable of stand-alone use, so is being released separately. It will be available on future versions of the DJ Bargain Bundle from Q-Celt Computing and available for download from websites and from QL PD libraries.

http://www.soft.net.uk/dj/software/freeware/freeware.html
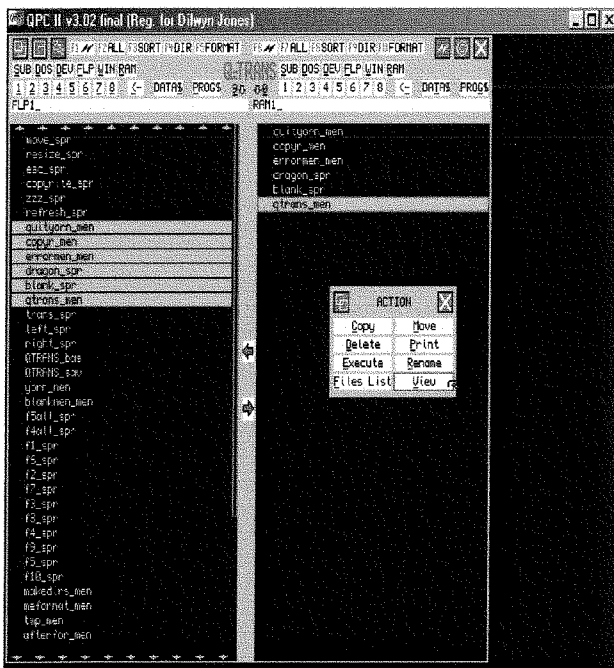
Q-Trans running in VGA mode

QL-Users List on the Move

The Internet Email Discussion list called QL-Users is now moving from its home at nvg.ntu.no. The move is due to anti-spamming measures being taken at the hosting site which has prevented people posting replies to the emails sent to the list. The list will now be hosted on Quanta's domain at quanta.org.uk. To subscribe to the list send a message to: **majordomo@www.quanta.org.uk** with '**subscribe ql-users**' in the content of the message, details on how to post messages will then be sent back to you. If you have any problems with subscribing please contact **bruce@q-v-d.demon.co.uk**. The mailing list is open to all not just to Quanta members and provides a lively discussion forum on QL/SMS issues.

SuperHermes Manuals

Tony Firshman was kind enough to give Phoebus Dokos all the superHermes Lite manuals in electronic form, so they could be posted online.

Two versions will be posted for each manual. One in printable form that makes a A5 sized booklet and one in the regular (in-sequence) form.

A replacement Aurora manual is also available from the same website.

These will obviously be very helpful to anyone who has either lost their manuals or purchased a second user unit without manuals. This is becoming quite widespread in the outside world, making replacement manuals available online and is a development to be welcomed.

The manuals can be downloaded from:

http://www.dokos-gr.net/~phoebus/

A reminder that replacement manuals for Miracle Systems Ltd products (Gold Card etc.) are also available from the Dilwyn Jones website.

Dilwyn Jones PD Library News

Version 1.20 of the QL Emulators CD will be available by the time you read this. It will feature the most recent version of uQLx emulator and updates to some of the most popular free software bundled on the CD such as Turbo and Editor.

The PD library has grown to some 150 packages of QL software in total (some packages require more than one disk) and still expanding fast. The games section for example consists of 15 disks, but a total of about 110 game packages.

The catalogue disk is available for download from my website, or send me an email to request it by email as a text file, or it can be obtained just by sending me a formatted floppy disk (or a blank CDR if preferred). Some of the software is available for free download from my website, but I've just about run out of space there now to put many more packages on the site. It may be worth a visit to the Software pages on my site to see if the package you want is available for free download there first.

Finally, Phil "The Library" Jordan and I have exchanged CD-ROMs of our respective PD libraries, so that we can work towards merging the available programs to be available from both libraries. The collection of available free software for the QL is now huge and it takes us both quite a lot of our time just trying to keep it up to date. Phil's entire library collection for example requires two CDRs to hold the whole lot. When you consider that a CDR can hold half a gigabyte or more of data, that's a fair reflection on the amount of QL software available.

Gee Graphics! (on the QL?) - Part 28

H. L. Schaaf

"Even More Connecting the dots"......

This time we add the Convex Hull to the menu.

The convex hull is a subset of the Delaunay triangulation and is easier to show than explain. You can think of it as a boundary that just barely encloses all the points, or what you'd get with a rubber band streched around the 'outside' points.

Since we've already done the work of locating the hull as part of finding the Voronoi edges, we use the information to select Delaunay edges that belong to only one triangle and then show them.

Continuing on from GG#27, take the mixture of code51_bas, D2V_bas, and the PROcedures for angl_frm, dist_btwn, SWAP and CYC that we had, and then merge in the listing "AddHull_bas".

Some may have noticed a few qwirks if they tried to get 2 squares from the grid option of the input menu. Looks like another little nudge is needed, so line 8285 has been used to give a better result.

Also if you tried the random input with 42 as seed with anywhere from 23 to 54 points, you would have noticed a strange division of the regions, but probably only with SMSQ/E. The 3 points numbered 13, 15, and 23 on the left edge are almost collinear. The triangle (#35) formed is an extremely narrow sliver with one angle being very nearly 180 degrees, so perhaps we have pushed the QL too far? This may be due to the way the random number is generated, another fun area for ma-thematical investigations. Anyone know the algorithm used to give the random numbers?

Still on my to-do list is pulling the Gee Graphics series together for posting on Dilwyn Jones website, where you can download the listings instead of having to type them in.

I've added Steve Poole's Voronoi program to my to-do list as well.

Next time we hope to do the Gabriel Graph, another subset of the Delaunay triangulation.

Listing AddHull_bas

```
102 REMark AddHull_bas to go with GG#28
112 REMark Convex Hull added April 30, 2002  HL Schaaf
5800  PRINT #0;"Delaunay : [P]oints, [T]riangles" ;
      :INK #0, 4
5810  PRINT #0;"        Voronoi : [V]ertices,  [E]dges"
5820  INK #0, 2 : PRINT #0 ;" [R]egions,  [H]ull, ";
5830  REMark  PRINT #0;' [G]abriel, ';
5835  REMark  PRINT #0;' [M]inimum spanning tree,
      ';cst$;
5840  INK #0, 4 : PRINT #0;\" [ESC] to Exit, [C] to
      [C]lear screen,";
5972   IF ans$ == 'h' THEN INK 2 :ConvexHull :
       choose_options
5975 REMark  IF ans$ == 'g' THEN INK 2 :Gabriel :
     choose_options
5978 REMark  IF ans$ == 'm' THEN INK 2 :MinSpanTree :
     choose_options
8285 IF (pat$== 's'):P(p_n,1)= P(p_n,1)-nudge
9890 :
9900 DEFine PROCedure ConvexHull
9910  LOCal i
9920  REMark choose the edges with only one triangle
9930  FOR i = 1 TO DIMN(Dedg)
9940   IF Dedg(i,0)=1 :show_edge(i)
9950  END FOR i
9960 END DEFine ConvexHull
9970 :
9980 DEFine PROCedure show_edge(edge_number)
9990 POINT P(Dedg(edge_number,1),1)
     ,P(Dedg(edge_number,1),2)
10000 LINE TO P(Dedg(edge_number,2),1)
     ,P(Dedg(edge_number,2),2)
10010 END DEFine show_edge
10020 :
10030 REMark end of listing AddHull_bas
```

RWAP SOFTWARE

QL Cash Trader v3.7 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT returns, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of the current tax year rules.

Q-Help v1.06 £10
Q-Index v1.05 £5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.
The PD toolkits referred to are available for £2.
Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and find out the commands which relate. Launch Q-Help for further info on the chosen command.

Sidewriter v1.08 £10

Produce landscape printouts of Easel/Qspread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix to laser printers. A most useful utility by Dilwyn Jones - you know it must be easy to use.

ProForma ESC/P2 Drivers v1.04 £8

New improved colour and monochrome printer drivers, providing up to 720dpi for all progr_ written for use with ProWesS, such as LineDesi_ Paragraph. Works on all Epson inkjet print_ support binary mode compression (740, 8_ models at least). 1440 dpi to follow.

QL Genealogist v3.26
Genealogy For Windo_

Store your family tree _
with details of their p_
those links build u_
Text files and _
individuals a_ _layout.
the perfec_ _making this
QL versi_ _of your family.
as a_ _nd QMenu as well
Samp_ _le and female trees.
PC ver_ _y since 1066 included.
appear _ - enter the details as they
these. QL _ _d it generates the tree from
PC version. _ _DCOM can be transferred to the
Both program_ _asy to use and complete with a step by
step tutorial.
** QL USERS upgrade to PC version for £25 ONLY **

D-Day MKII v3.04 £10
Grey Wolf v1.8 £8
War In The East MKII v1.24
(Upgrade Only) £5

Image D v1.03 £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

SBASIC/SuperBASIC Reference Manual £40
Updates £6 each, £10 for 2 (Current Version - Rel 4)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.
This book is ideal for all QL users and is kept up to date with regular updates.
Orders are currently being taken for the next print run of this popular tome.
(Note: Price for the book does not include post & packing).

QL Cosmos v2.04 £5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the progra_ for you - generates pictures of the stars and planets fo_ given place or time and provides details on these _ Includes Halley's Comet, the Moon and the Sol_ planets.

Q-Route v2.00
Upgrade from v1.xx

The latest version of this popu_
Find the quickest route or th_
two places, using roads. A_ _able
for this program (see el_ _rogram
is easy and quick to _ _wn places
and roads to the _

Flashbac_ _only) £5

The ultim_ _nely fast and flexible,
easy to _ _he latest versions of the
QL o_ _ntained. A report module is
in_ _t output in any way, including
_only available as an upgrade from
_riginal still available from Sector

_den v3.08	£10
_ MKII v2.03	£8
_rawn v2.01	£8
_rrorday v3.1	£8
_West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

_ll of A wealth of QL adventures - mainly text only.
Save the Galaxy from the ambitions of the evil dictator Nemesis.
Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horrorday.
Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.
Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.
Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.
Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.
All six adventures are available together for only £25.

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while.
Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

RWAP Software, 7 Common Road,
Kinsley, Pontefract, West Yorkshire
WF9 5JR
TEL: 01977 614299

* Also known as Trading Accounts
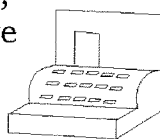
Cheques in £sterling
payable to 'R.Mellor'

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for? All 6 programs only £28.

£8
£5
£5
£5
£10

_f v5.20
_ll v2.07
_II v2.00
_er II v1.2
_ne v1.2
_strike v1.5
_htdeck v1.05

The latest maps for Q-Route. Maps of areas of Britain have been created by them out of Big Britain Map - they will use memory and can contain more detail. All covered: Scotland, NE England, NW England, S&W Yorkshire, Wales & Derbyshire, London area and South England.
Latest version of Q-Route is recommended.

£2
£5
£2 ea.
£5
£2
£2

Britain.map v1.11
BIG Britain Map (needs 2MB) v2.03
Various Britain Area Maps (ask for details)
Ireland Map v1.00
Belgium Map v1.01
Catalonia Map v1.03

The QUANTA Workshop and AGM in Manchester

Dilwyn Jones

On Saturday 13th and 14th April, the local Quanta sub-group in Manchester, NEMQLUG, hosted a Quanta workshop on the saturday and the Annual General Meeting on the sunday. Attendance was pretty good on the saturday but rather poor on the sunday, so this may mean the end of two day workshops in England anyway, apart from special international events like the QL2000 meeting.

to allow the southern traders time to travel north on the saturday morning – would the same arrangements have been made for southern workshops I wonder?

The venue was the local Scouts hall, a very short distance from the M60 motorway, so very easy to find, an important consideration when you have a sense of direction like mine!

Inside the hall early on the Saturday

The saturday workshop was timed to start at mid day, which was a bit of a departure from the norm, as the intention was

From the first moment, it became apparent that there was a significant number of 'men in black' (well, men and women in black) about. Turned out to be the staff of D & D Systems along with Peter and Claus Graf and a few assorted Q60s which everyone was drooling over all

day (the computers anyway, don't know about the staff).

D & D Systems have put together quite a bundle of well known QL software to enhance the Q60 experience. The Q60 comes with a sumptuous manual as well, once you inherit a Q60 you have a lot of getting used to the sheer power and variety available to you!

From the largest presence at the show to the smallest. Keith Mitchell had one of the smallest 'QL' systems I'd ever seen. Running multiple copies of the uQLx emulator, this tiny little laptop was quite an attraction in itself.

Surprise appearance at the show was by former QL trader David Batty, who was one of the leading QL software traders in the 1980s. David has been keeping a close eye on the QL scene via the internet and has lately been taking an interest in the Turbo compiler source code due to his interest in compilers generally. He still has several QLs, he said, and took away with him a copy of the QL Emulators CD so we may be hearing more from him in the future!

Another pair of suspicious characters were lurking about the show, taking video pictures of anything that moved and quite a lot that didn't. Turned out to

Keith Mitchell's little laptop running multiple uQLxes – note the size compared to the little biro above the keyboard

Keith Mitchell with his hands in a QL of some sort as usual
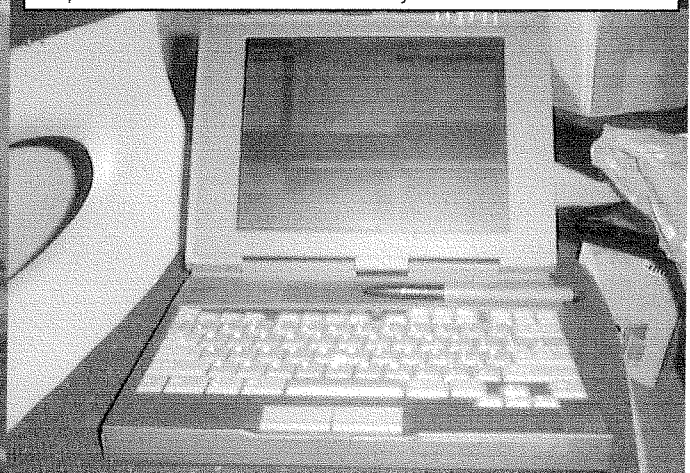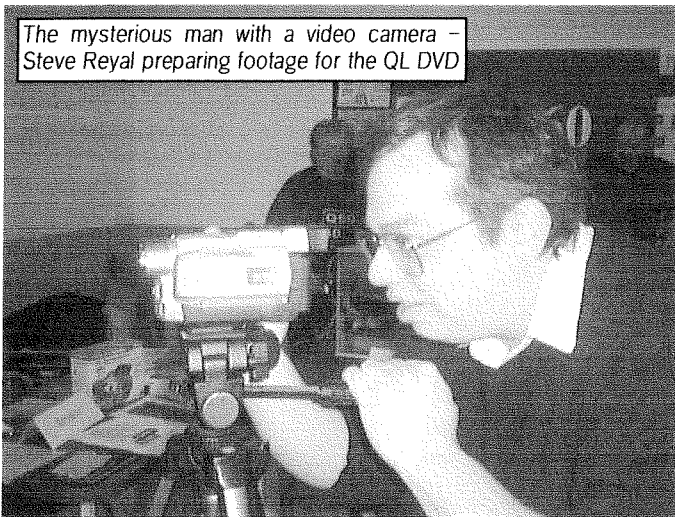
The mysterious man with a video camera – Steve Reyal preparing footage for the QL DVD

Bernd Reinhardt (he refused to write an article so we print his picture as a punishment!)

be Steve Reyal and Darren Branagh, both busily preparing material for the QL DVD, a promotional video about the QL scene in 2002. They had produced a short trailer for this on floppy disk (some PC video format), recorded and processed to look like one of those scratchy newsreel films. Rather corny, but attention grabbing all the same. I look forward to seeing the finished version – hopefully it will be good 'propaganda' for the QL scene.

Jochen Merz, Tony Firshman, Geoff Wicks, Q-Celt Computing, Bill Richardson and Roy Wood of QBranch had all made

the trip to the show, but Rich Mellor of RWAP Software was again unable to attend due to ongoing illness. He is making a slow recovery, but not yet well enough to travel long distances. However, as readers of the QL-users email mailing list will know, he is still actively programming and releasing new software and updating other programs. Most of his products, including the excellent updated Q-Route v2.0 route finding software, was available from other sources at the show.

Geoff Wicks has been dropping some hints recently about a new product from Just Words. His adverts have carried the question-marked black briefcase, but despite repeated attempts to extract the information from him, a quiet smile was all I got, so I guess I'll have to wait until his QL Today news items come in before I'll find out what it is.

Jochen Merz and Bernd Reinhardt manned the JMS stand. Jochen was selling the latest release of QPC2 which among its newer tricks can now BEEP from the host PC's speakers, rather than the

rather pathetic little system speaker. This is more important than it may seem – Windows NT users had problems getting audio out of the system speaker before, so now you can hear the QL beep as loud as you want (or can tolerate!). Despite this, QPC2 has some way to go before it can match the 20kHz stereo audio from a Q60 for example, or a QDOS Classic system equipped with the QL Sampled Sound System (QLSSS)

David Batty of Sector Software
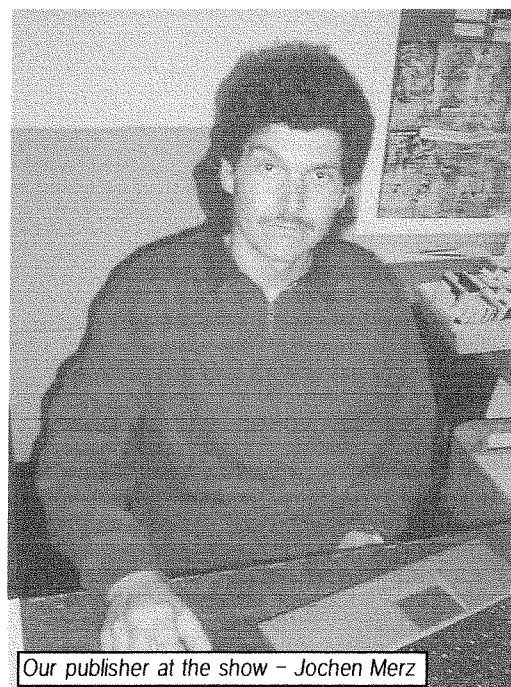
Our publisher at the show – Jochen Merz

A well stocked 'bring and buy' stand (pair of stands actually!) meant there was plenty of older QL goodies to browse through and snap up at bargain prices.

Q-Celt Computing and I claimed the record for the shortest time ever for a QL CD-ROM to go from concept to production. The QL Documentation CD went from an idea Darren and I had over a beer the night before the show to being sold the following day – it took us just 3 hours to produce the night before the show. It brings together into one collection the publicly available text files and documentation and articles about the QL, its hardware and software and programming information. The idea is that it will act as a single source of freely available documentation, which we hope to continuously update as more material is contributed. Costing just 5 pounds (plus postage if ordered by mail) this CD proved to be a good seller at the show and

several people promised to send more material for it over the following weeks. The CD itself is freeware and may be freely copied subject to any restrictions imposed by individual authors of the material supplied, for example, since a text file copy of the QL manual is included, it would be necessary to obtain permission from the rights holders in north America (Paul Holmgren or Frank Davies) to use the CD there.

A "social event" (dinner) was held on the saturday night. Sarah Gilpin had organised this function by taking advance bookings from those interested in attending. I did not attend myself, so cannot report on the

Bill Newell and John Gilpin (right) of Quanta

normally lengthy discussions on QL matters which take place at these "social events", but thank you Sarah for organising this function.

The AGM itself on the Sunday proved to be an uncontroversial affair. Long serving treasurer John Taylor is retiring from the post this year and his place will be taken by John Gilpin. John Taylor was thanked by all present for his service. A minor error on the paperwork issued with the newsletters in advance of the AGM meant that there was some uncertainty over the position of one of the officers, but after a short discussion the AGM concluded there need be no problem after all. A discussion was also held as to who should be appointed auditor this year, since the smaller membership of the past couple of years meant that there was less work for an auditor and some savings could probably be made on auditing fees.

Q60 Droolers Corner

Well, you can drool over the computer, I'm not so sure about the staff. There was so much QL-style black on this stand, I hope the pictures won't be too dark! Anyway, here's some pictures of the Q60 itself along with the men behind the Q60 and D & D systems.
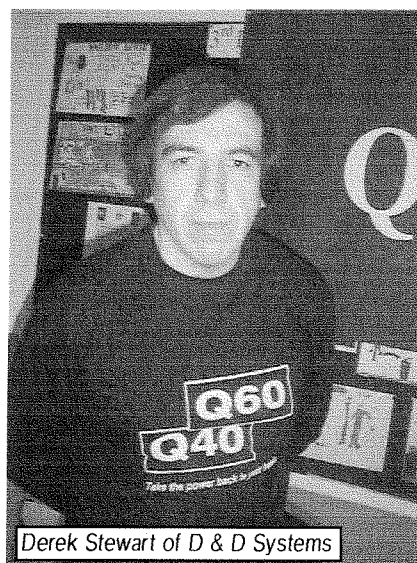
Derek Stewart of D & D Systems
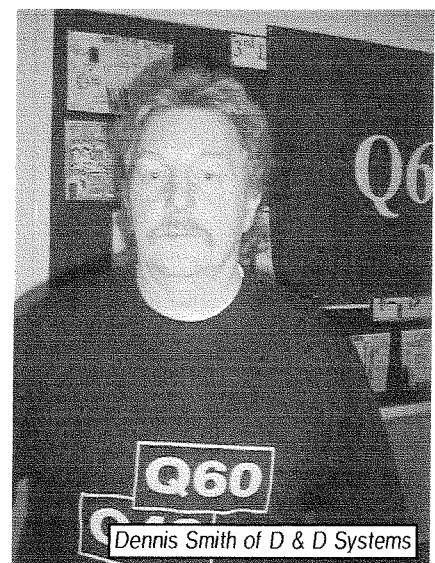
Dennis Smith of D & D Systems
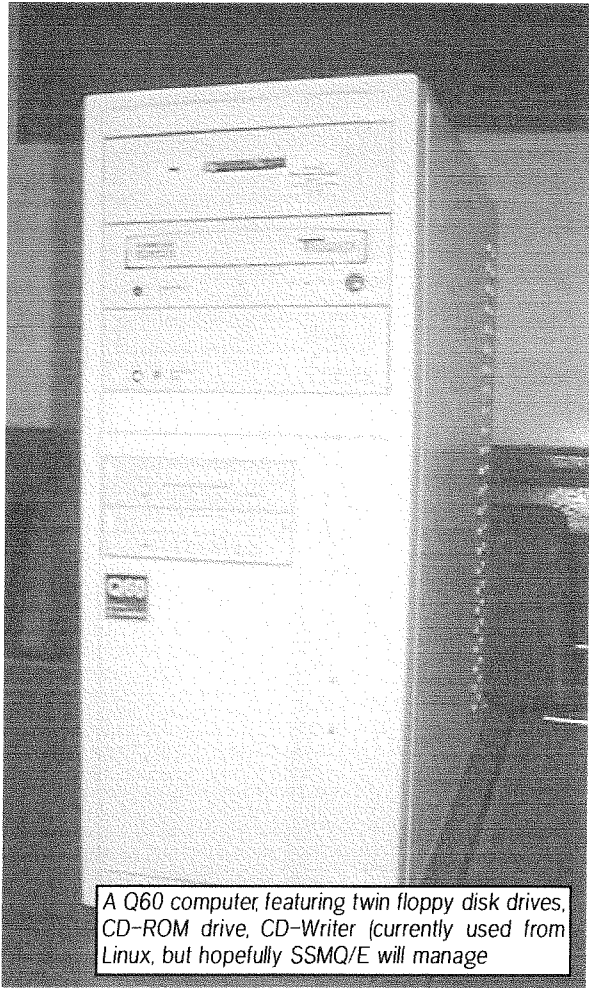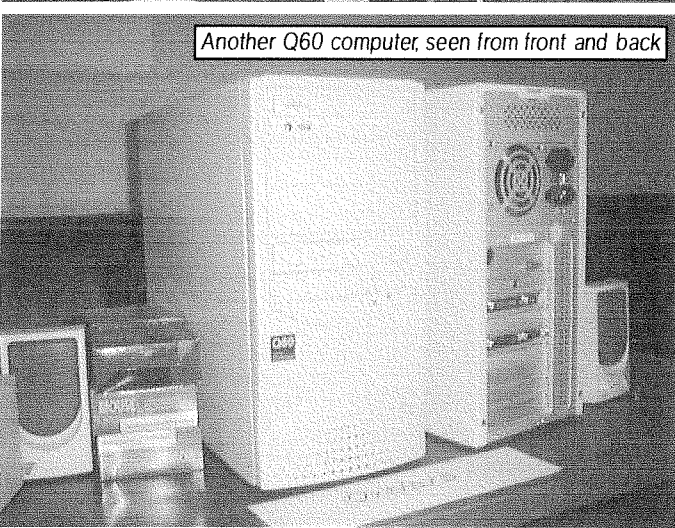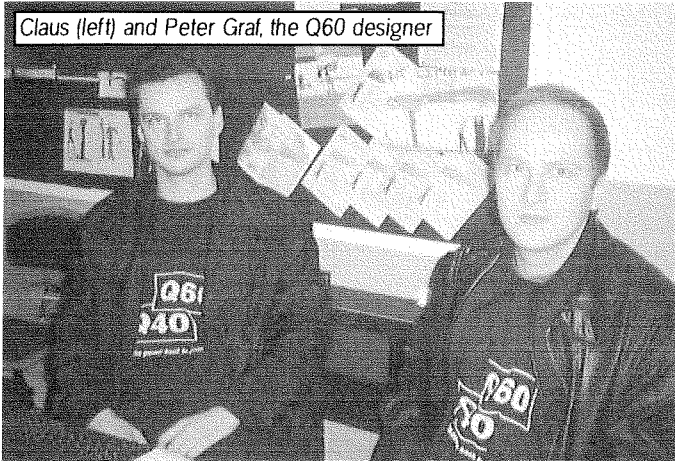
Claus (left) and Peter Graf, the Q60 designer

Another Q60 computer, seen from front and back

A Q60 computer, featuring twin floppy disk drives, CD-ROM drive, CD-Writer (currently used from Linux, but hopefully SSMQ/E will manage

Pointer Programs

John Perry

QL Today is often quick to promote the virtues of pointer environment and SMSQ/E, hardly surprising perhaps since the publisher sells SMSQ/E. I ventured to ask the editor why, given the amount of PD software there is for the QL, so few reviews of the ever increasing number of such pointer driven programs were published in QL Today. His reply was brief and to the point: "Nobody writes any for us - why don't you?"

So, I did. I duly obtained some disks of QL PD software and decided to try writing some of my views about these programs. You may not agree with what I write, but this is what I think of them. Here's my first review.

SPELLING-CRIB v2.10
by Geoff Wicks

This freeware program is from the highly respected Just Words stable and it shows. It's not a true spelling checker as it's name implies, but rather a utility for helping to find the correct spelling of difficult words.

This version relies on the presence of QTYP, Tony Tebby's pointer driven spelling checker program. QTYP is a long established program, capable of checking the spelling of words as you type or checking entire files. QTYP is a commercial program and rather a heavyweight

one at that - it is not the easiest of programs to set up and use. If you do not have QTYP you will not be able to use this version of Spelling-Crib. Rather, you should obtain a copy of version 1.00 from the author, which instead makes use of plain text file dictionaries. I have not seen that version, so can't comment.

QTYP dictionaries are sizeable ones (the one supplied with Spelling-Crib has about 65,000 words and is about 120KB long, dwarfing Spelling-Crib itself which is only about 40KB) and may be edited by the user using the QTYP dictionary editor. Over the years, several additional dictionaries have been created by users worldwide, so there should be a good choice of dictionaries available - only one English

dictionary is supplied with Spelling-Crib, though if you have another QTYP dictionary it should be possible to use that, as Spelling-Crib allows you to configure the path and filename of the dictionary. The same configuration makes it possible to make it run from a directory on a hard disk. The copy I got from a PD library was ready configured to run from FLP1_.

It is also possible to tell the program where to find the dictionary by passing its filename to Spelling-Crib as a parameter in an EX command:
EX WIN1_CRIB_SPELLCRIB_OBJ;
"WIN1_CRIB_QTYP_DICTIONARY"

The Help screen
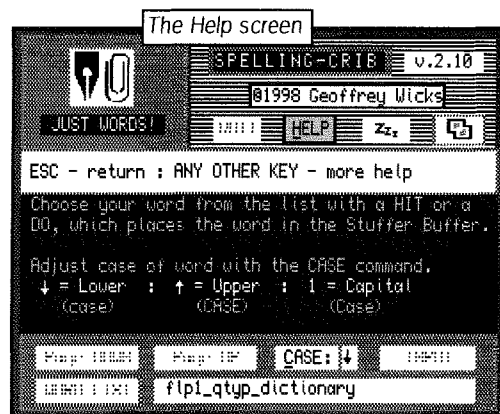
As might be expected of Geoff Wicks, the manual is a total pleasure to read. All the information you need, including system requirements and a 'Quick Start' section are there. The manual tells you everything you need to know without waffling on at great length. The manual is a straightforward Quill DOC file. Beware though, there is a separate UPDATES_DOC file which you should also read. The manual warns that users who have other programs making use of QTYP (e.g. the Text 87 spell checker) may have programs as apparently only one program at a time can use QTYP. As I don't have Text 87, I could not test this.

Spelling-Crib opening screen
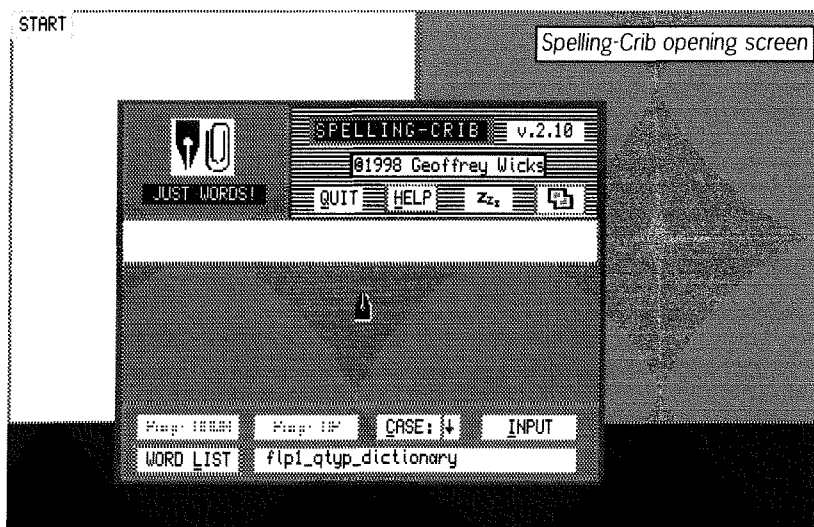
At 40KB the program itself is probably so short that no version without linked compiler runtimes is necessary. For those unfamiliar with this, QLiberator compiled programs can come in two forms, one for people who already have the compiler and another for those who don't. A little library file about 10KB long called QLIB_RUN (QLiberator Runtimes) is needed for QLiberator compiled programs to run. This can be compiled into the program so that the program can run stand-alone with no need for the compiler to be present. The snag with this though is that if you run several such programs each having runtimes linked, you are essentially wasting 10KB of memory each time. Some software authors provide versions of their programs which don't have these runtime libraries built in, so if you have the compiler itself on your system you can use slightly shorter versions of these programs. Sadly, Spelling-Crib does not provide this feature, you get the one version with linked compiler runtimes and that's it. At least it makes life simple that way and frees you from worrying if your version of QLiberator is modern enough for this program. And in these days of large memory QL systems we're probably not going to worry too much about a program being 10KB longer. The program is pretty well standard Just Words black and white – if you've seen any Geoff Wicks pointer driven programs they tend to look pretty alike in terms of screen appearance, he seems to have developed his own in-house style. To search for a word, you need to know what letter it starts

with. Unknown parts of words can be represented by a forward slash character. So if you didn't know how to spell "character" you could click on the

Display of words found after searching for a/b/c
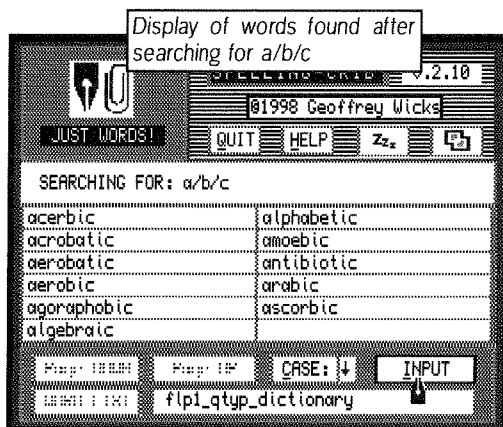
INPUT item (see the screen dumps to see how the display looks) and enter ch/er – the program would then list all words it knew which started with ch and ender with er – quite a few of them!
The program can go a step further than this. You can specify more than one 'wildcard' if required. Try entering some-

Entry of search word

thing like a/b/c. I didn't think this would list many words, but actually it did manage to find quite a few. Again, see the screen dumps for an example. If the program finds more than a screenful of matching words, there is a little item you can click on called PAGE DOWN which will step through the list. When you reach the end of the list, the writing in that button becomes hard to read and it becomes no longer available,

so indicating you have reached the end of the list. Similarly there is a PAGE UP button, to allow you to go back through the list if required.
The use of "/" as a wildcard seems a bit unusual – I am used to entering a "*" as a wildcard of any number of letters, or "?" to represent a single unknown letter. Wild-cards are symbols which represent unknown letters, so a/d could be "and", "abandoned", or "abated". You get the idea. Perhaps the reason in this case is that Spelling-Crib does not need to make the distinction between single unknown characters and groups of letters – the "/" can represent one or any number of characters.
Don't try to start a search with a wildcard character, it won't work. You have to know the first letter of a word. Therefore, a/ will work but /a will not work. Trying to be clever, I attempted to list the entire dictionary by just entering a single / character or a double //. The program just complained 'unsuitable entry' and asked for another entry.
It is possible to step through the dictionary one letter group at a time though, by entering a/ or b/ and so on. It surprised me to find that the dictionary started with half a dozen entries all starting a double a. The dictionary includes some proper nouns as well as the usual dictionary entries.
When entering a word, the program does not allow you to ESC out of a new entry, or even make a blank entry, in order to go back to the previous listing. Not a serious problem at all – you just enter the same wildcard again.

The program has a short built in HELP facility, enough to use the program without reverting to the manual every time you need to use Spelling-Crib. To wade through the help text, you have to press a key on the keyboard – PAGE DOWN and PAGE UP are not active. You can, though, press the right mouse button to step through, presumably this just acts like an ENTER keypress. Similarly, when you use the QUIT button, it asks for Yes/No confirmation via keypresses, no option to respond using a mouse.

Spelling-Crib has the usual MOVE and SLEEP icons. If you click on Zzz and you have Qpac 2 installed, the program goes to sleep as a 'button' in the button frame. The double-square move item lets you move the program's display to another position on the screen. As the program uses less screen area than the QL 512x256 screen this is quite useful – especially when using larger sized displays. The program can be parked at the most convenient part of the screen.

The program runs either in QDOS with pointer environment or in SMSQ/E. In both cases, QTYP must be present (obtainable from QBranch for about 30 pounds), and the system must have Toolkit 2 (available on most modern QL systems and interfaces, including SMSQ/E).

Spelling Crib is freeware and should cost no more than about a pound from most sources of PD software or direct from the author himself. The review copy came on disk GE05 from the Dilwyn Jones PD Library Service. Just a pound for a program of this quality is a real bargain.

System Set
by Dilwyn Jones

System Settings is an uninspiringly named program to provide facilities to, umm, change system settings. It's a pointer driven program and although it sets out in what it achieves to do is not the most exciting of programs and does not seem to have been that well publicised – I stumbled across it rather than set out to find it. Although I am quite glad to have found it in the end.

Like most of the author's offerings, this one is easy enough to use and has a few quite good points, but is not exactly a program which stands out from the crowd.

The program needs pointer environment and Toolkit 2. There are 5 broad functions of this program.

1. Clock setting – time and date
2. Device names – renaming the device drivers
3. System defaults – setting DATA_USE etc
4. Directory creation
5. File selection and starting

If you have ever used system setting software in other operating systems such as the Windows Control Panel, you'll know how fiddly and awkward they can be, especially if you're not sure what you are doing and object to having half a ton of paper on your lap. Thankfully, System Settings is quite easy to use and on the whole you could probably get away with no manual, as long as you understand the use of DEV_USE commands and the like.

The first allows you to read and set the system clock. While using SDATE from BASIC is not exactly a pain, it does mean you have to remember the order of the parameters and

check if the clock is correct. System Settings gives you a box (see the screen dump) with 6 smaller boxes and 'READ
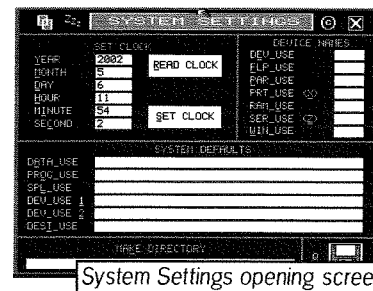
System Settings opening screen

CLOCK' and 'SET CLOCK' items. Read Clock displays the current time and date. Clicking on Year, Month, Day, Hour, Minute, or Second boxes lets you type in a value for any which are not correct.

Once entered, simply click on SET CLOCK and the clock time will be updated. Nice and simple, the only complain I would make is that perhaps a continuously updated clock display (the Toolkit 2 CLOCK command for example) could be shown as more than once I found that by the time I'd clicked on SET CLOCK I was a second or two out afterwards because I'd clicked too soon or just too late. The display of this program is a little cluttered though due to the sheer number of items it is able to alter, so perhaps the distraction of a constantly changing clock display would just add to the clutter, and you could simply start one of those ubiquitous little clock programs running while using System Settings.

The next section lets you alter the device names on your system. This facility is just like FLP_USE, DEV_USE and such commands. Indeed, being compiled BASIC, this program presumably makes use of those extensions.

You can set new 3 letter names for the FLP, DEV, PAR, PRT, RAM, SER and WIN devices. As long as the equi-

valent xxx_USE commands are present on your system presumably. Not all systems include a WIN_USE command or SER_USE command so these may not be useable on some systems.

The third section lets you alter what it calls 'system defaults.' This consists of data default device (DATA_USE), program default device (PROG_USE), SPL_USE, DEV settings for DEV1 and DEV2 and the desination default (DEST_USE). Not a lot to say about these except that they allow you a good length of input and seem to work exactly as described.

The Make Directory box simply lets you type in the name of a directory and creates that directory. Avoids the need for MAKE_DIR commands from BASIC. Obviously, will only work on systems which support making such directories (Level 2 or higher filing systems), not on very old disk interfaces. I think the Trump Card was the first disk interface to support these (open to correction on this!)

Make Directory appears to offer no advantage over a MAKE_DIR command until you explore the final option, the 'floppy disk icon' box. Click on the red and white floppy disk and up pops a rather nifty file display box from which you can display lists of files on given drives via single key-presses or mouse clicks. Along the top, predefined buttons appear for CDR (CD-ROM?), DOS, FLP MDV, RAM, ROM and WIN devices, with further but-
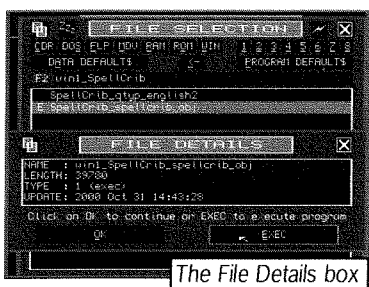
The File Details box

tons for drive numbers 1 to 8. These appear to be hard coded into the program and are always shown even though the device is not present on a system, e.g. I have no CDR device but CDR is available. Also if device names are changed these don't update.

So to view files on FLP2_ just click on FLP and on 2.

The <- button acts rather like a similar one in QPAC2 in that it goes back one level of directories. Imagine you had directories set up as follows. WIN1_ contains a directory called Psion, which in turn contains a directory called Quill which itself contains a sub-directory called docs. If you were displaying the list of fiels from WIN1_PSIONS_QUILL_DOCS_ and clicked on the <-button, it would take you back up to WIN1_PSIONS_QUILL_

In the list of files itself, each
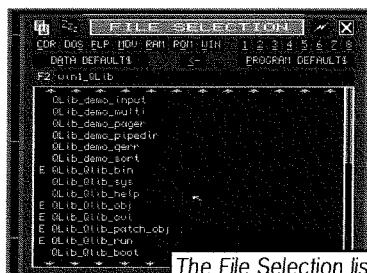
The File Selection list

filename is preceded by a character which indicates the file type. An executable program (one you can EXEC) has an E before it. A directory has a > symbol in front of it, and although I don't know what an S-ROFF file is, apparently these are preceded by a letter R. If you click on a filename preceded by a > the program goes on to show a list of fiels in that directory. So you can browse through directories on a hard disk or ED disk by clicking on > to enter that directory, or <- to go back. Quick and convenient.

If you left click (HIT in QL termi-

nology) on a filename, the program's next trick is to stuff that filename into the stuffer buffer. If you then go into another program and press ALT SPACE when it asks for a filename, the filename is "typed" into that program. In effect, this program can act as a pointer driven filename selector for any program, even non pointer driven ones like Quill!

That would have been useful in itself, but an extra surprise awaits. Right click on a filename and a box pops up with brief details on the file – its length, full path and filename, the file type and the date last updated. Click on OK to clear this box. If the file was an executable one, you can even execute it by clicking on the EXEC button. A further menu appears allowing you a wide range of execution options. Memory grabbing programs like Quill can be tamed by
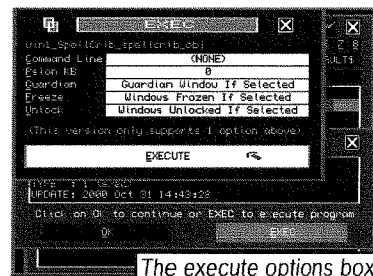
The execute options box

specifying how much memory allowed, like the P Psion option in QPAC2 hotkey commands. Guardian windows, freeze (for programs which don't suspend when you CTRL C out of them) and Unlock Windows (for programs like clocks which need to write to the display all the time irrespective of whether they are the currently active program or not). A command line can also be passed to the program, e.g. if starting an editor and the program allows you to pass the filename of the file to be entered, some programs also allow their default drive details to be passed

in this way (e.g. Spelling–Crib!) Mostly, you will not be using these options, and will just click on EXECUTE to start the program.

It appears that only programs can be executed. System Settings does not seem to be aware of FileInfo II, which would have allowed you to execute a data file and the system would fire up the program associated with it.

Having used the DEV and other settings before executing a file, you can even use this program as a limited program launcher for older awkward programs like Quill which aren't aware of hard disk directories and have habits like gobbling up available memory when they start.

The program has the usual Move and Sleep items represented by the double square symbol to move the program around the screen and the Zzz symbol to make the program go to sleep in the button frame. The Copyright symbol brings

System Settings copyright screen, including that dragon!
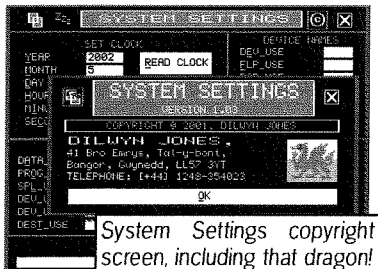
up a little copyright screen,

which displays the author's contact details and a cute little Welsh dragon. Incidentally, try clicking on the dragon to see what it says (discovered that one by accident, seems to be a little hidden feature)

Leaving the program is by clicking on a little [X] icon in the top right hand corner and you get a little confirmation Yes/No box to check if you really wanted to exit, although with a program of this kind where there is no data to save, it's not strictly necessary and it would be nice to have an option to configure this to be disabled.

As it stands, you can configure just the file selection menu. You can specify the drive it first shows and whih directory, plus the number of fiels the list is able to show.

A curious "feature" of this program in earlier versions was that if a drive had just one file on it, the program failed to show that file. Correspondence with the author revealed this to be down to a problem getting menus in the programming tool he uses (Easyptr) to display menus with just one entry. Just before sending off the first version of this review, he sent me an updated version 1.03 which seems to fix this problem.

The program comes with a Quill DOC file manual which is

adequate but not outstanding. I suppose it's not easy to write a good interesting manual for this kind of program.

It is hard to either put down this program or praise it. It sets out to do something and does it, although seems to have lost its way a little on the way. A System Settings program which also has a file selection utility? Doesn't quite seem to fit the bill somehow.

Also, I would have liked a 'System Settings' program to offer additional facilities like setting mouse speed, keyboard auto-repeat settings and the like. Also, I'd like to see the devices list contain only devices which exist on the QL system it's being used on. While I'd hardly urge it to be a Windows Control Panel for the QL, it does manage to be useful and distinctively QL rather than just like any other system settings utility on other computers. In summary, I'd say this is a good program, one point away from making the move from a good program to a very good one.

Not everyone will want a program like this, but for anyone who does, this will fit the bill quite well. It seems robust and easy to use without being confusingly full of facilities you'd never use half of.

Kodak Digital Camera Driver for Qdos

by Simon N Goodwin

This article explains how I have made it possible to use a Kodak DC-series Digital Camera from a Qdos or compatible system. The driver works with several models of camera and many Qdos-compatible systems - the only thing you really must have is a serial port capable of reading a block of a thousand or so bytes at a time without stopping.

This work was first demonstrated at QL2000 and has since been tested by eminent Qdos users including Darren Branagh, Claus Graf, George Gwilt, Phil Stokes and Per Witte. It is known to work with SuperHermes, Qdos Classic on any Amiga or Q40 or Q60, UQLX, and SMS on Atari ST, TT, Q40 and Q60. It does not work with the 8049 QL IPC or Amiga Qdos because the camera output over-runs the serial input buffer.

The article also explains how digital cameras work and what they can do, and why some work with Qdos and some will not, with specific and general tips on choosing compatible ones. The driver is freely available on Dilwyn Jones's web site, at http://www.soft.net.uk/dj/software

Digital Camera concepts

Digital cameras all do much the same thing. They are portable devices that take and store pictures and transfer compressed images to computers. The greatest variation comes in the way they store and transfer the images - there is no standard for the messages that pass between the camera and computer. This Qdos driver is based on information provided by Kodak, which is why it works with some of their cameras, and not others. The choice of Kodak was determined by their policy of making protocol details available, unlike most vendors, and because they make capable cameras that are readily available.

The cameras have various internal settings, for things like the time, flash, zoom and image quality, such as resolution in pixels and degree of compression). These are stored in tables internally - one table gives the current settings, while another is attached to each image, storing the settings when that picture was taken. The digital camera driver allows access to these tables, as well as commands to make the camera do things and access to the stored photos.

Raw image data is usually compressed in JPEG format. This familiar open standard was devised by the Joint Photographic Experts Group and does an excellent job of packing photographic images to a tenth, or less, of the space they'd otherwise need, without obvious degradation. There are several QL programs to decode JPEGs, of which Dave Westbury's Photon is my favourite as it's small, very compatible and delivers fine results even on a standard QL screen.

Top resolution of the typical 'megapixel' cameras supported by this driver is 1152 by 864 pixels, so Photon scrolls those or larger JPEGs across the QL screen. Each JPEG image delivers a 24 bit colour value for each pixel, comprising eight bit Red, Green and Blue components. Internally most of the currently-supported cameras use a 1160 by 872 point CCD or 'Charge Coupled Device' sensor array. Since the resolution is part of the image data, the same driver can read images with more or fewer pixels.

DIY digicam

The CCD is essentially a dynamic memory with the lid off. Twenty years ago hobbyists made low-resolution cameras by prising the lids off 4116 (16 kilobit) DRAM chips and focusing light on the exposed surfaces. Each bit in a DRAM is represented by an electrical charge, which leaks away unless it is regularly re-written - hence the need for such 'dynamic' memory to be 'refreshed' many times a second, to ensure that it doesn't forget your data.

Like the EPROMs used in the QL cartridge port, RAM happens to be light-sensitive, so the intensity of the light falling on each part of the memory grid determines the speed with which the charge is lost. Normally the chip is covered, but if you uncover it you can take pictures by writing the same value to each bit and waiting for it to decay to the opposite state. The longer this takes, the darker the corresponding point in the original image. Within milliseconds you've got a grey-scale picture.

Modern digital cameras work the same way, but offer more than the few thousand bits possible with a 4116 chip, a more regular array, and filters to distinguish colour. The pattern of the filters is a grid, alternating by row and column, known as the 'Bayer' or 'CFA' pattern. This is tailored for JPEG compression and the vagaries of human sight.

The first row of sensors responds alternately to red and green pixels. The next row responds to green and blue, and so on for odd and even rows and columns. The intensity of white light can be measured for every pixel, but the exact colour can only be known for groups of four.

The basic pattern is square, two pixels on a side, rather than rectangular. There are twice as many green points as there are red or blue, matching the sensitivity of human eyes - to mix RGB components to grey scale, you need more green than of all the other components or the mixture has a purplish tint.

Most of the images I collect are taken at the VGA resolution of 640 by 480 pixels, which the camera supports as an option; this is fine for web pages and relatively quick to store and load. I only use higher resolutions for images that I want to edit later or submit for printing in glossy magazines.

Even modern digital cameras offering millions of pixels barely approach the resolution of 35 millimetre film, but unless you are an expert photographer using a tripod, expensive lenses and a good lab the practical limit will be your expertise, not

the technology; even heavily compressed VGA resolution images can yield results that look as good as compact camera prints to a non-expert, if you get them printed on the same paper. Boots and some supermarket developing services can now print from memory cards as easily as they can from photographic film. Results are better than if you print the images at home, though Phil Stokes has done quite well with Photon and his Qdos-driven inkjet.

JPEG deliberately exploits the fact that human eyes resolve more accurately in grey-scale than they can in colour. The first step in compressing an image, as with the video standard MPEG, is to divide the data into colour and brightness information.

Colour data is sampled at half the rate, vertically and horizontally, of that used to measure brightness. Only one colour value need be processed for every four brightnesses. The lower spatial resolution of the eye for colour means that humans don't notice the missing data, so we get away with storing four brightness values and one colour set, rather than twelve values - four each for Red, Green and Blue.

The single colour value could be encoded as an RGB triple but that would duplicate some of the brightness information, so it's more useful to convert the RGB into the equivalent HSV (Hue, Saturation and Value) or YUV (colour difference) triple. These are other ways of expressing colour in three parts, but here one corresponds to the brightness already measured for each component points, so we can discard that and end up with six numbers - two for colour and four for brightness.

Raw colour images would require twelve numbers, consisting of four triples, so we've achieved a 50 per cent compression and a better match with the physical properties of the eye, the CCD, the LCD display and the JPEG standard.

Compression factors

Other stages of JPEG compression depend on the original data and how aggressive the compression is. JPEG goes on to divide the picture into larger blocks, encoding these basic patterns and the differences between them. The variability comes from the way such differences are han-

dled. Smaller differences are discarded to reduce the size of the compressed data. The quality factor determines the trade-off between fidelity to the original and reduction in the file size.

These Kodak cameras offer three compression settings: good, better and best - like burger chains, 'small fries' are not on the menu. These terms describe heavy, medium or light compression, resulting in small, medium or large files. Even then, the actual size depends on the subject matter, and if there's only estimated to be room for one more picture, it might not be stored at all if it turns out to be unexpectedly hard to compress.

The disadvantage of packing is the need for CPU power on the camera, and processing time to pack each image, but this is more than outweighed by the space saving. My DC200+ can easily squeeze 50 images onto a 4M flash memory card; without compression there would only room for four, even at the lowest 640x480 pixel resolution in 24 bit colour: 307,200 pixels at three bytes each demand 900K per image. A 32Mb card now costs about £20 and can store 400 images - it only takes a few seconds to swap cards, and the contents are retained indefinitely even if the card is not plugged into the camera.
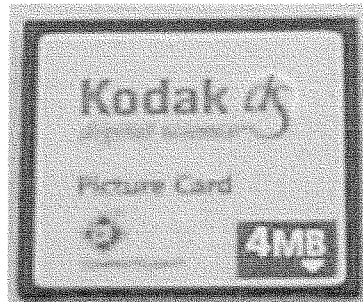
This 4 Mb Compact Flash card holds about 50 pictures in compressed JPEG format, 640 by 480 pixels in 24 bit colour.

We have an excellent free decoder for JPEG images, Photon by Dave Westbury. This works on all Qdos-compatible computers. It's obviously quite slow on a standard QL (taking minutes to decode a full-screen image) and limited in colours but the faster your computer and the more colours you can display, the better Photon performs.

What works

The driver is written in SuperBASIC with a couple of machine-code extensions that are not essential but help to speed it up. All the software is open-source. The main program allows users to download pictures from the camera and tell it to take and store a picture, with SuperBASIC com-

mands. George Gwilt is working on a menu-driven Turbo-compiled front-end for the driver, which will support QPTR without forcing its use, but the BASIC program is already quite usable by typing commands or stringing them together in scripts of your own.

Compatibility

The driver is known to work with Kodak DC200, DC200+, DC210, DC210+ and DC215 cameras. These were produced over the last couple of years and are quite easy to obtain, though they are no longer current models. The same software protocol is used by other Kodak cameras, with minor variations (synthetic names are used to access images, rather than internal index numbers) so the DC240 and DC280 and recent models like the DC3400 and DC5000 should also work.

The older DC50 and DC120 are also based on this protocol, and a few lines of BASIC should get them working too - if you have one of those and want details of the changes, please email me at

qdos@studio.co.uk

There is some support for them in the driver but it's not complete or tested as so far no one with one of those cameras has contacted me.

There are hundreds of models of digital cameras, and dozens of incompatible communication protocols used to link them to computers. I wrote a detailed comparison of the protocols, including some of the general information in this article, for the December 2001 issue of Linux Format magazine, on pages 49 to 55.

A lot of cameras use the so-called Universal Serial Bus or 'USB', and since there is no interface for this from Qdos-compatible systems they are physically incompatible with Qdos. The problem with interfacing USB with Qdos is that the complex chips that control USB from the computer end are all made for Intel's 32 bit PCI bus - even Apple Macs use those, but PCI has nothing in

This self-portrait was taken at the Quanta AGM, using Simon Goodwin's DC200+ to photograph the identical model Darren Branagh snapped up for £80 second-hand when he heard that it could be used from Qdos

common with current Qdos-compatibles and the hardware effort needed to support USB directly, let alone the software, would be very great (please prove me wrong, Nasta). However there are two interfaces that do work - RS-232 serial and Compact Flash.

The first thing to check is that the camera has an RS232 serial port - USB-only cameras are not going to work with Qdos, though if they use compact Flash storage you might be able to connect them via IDE. The part numbers matter as well as the brand, as cameras in each range may use Compact Flash, MMC, or Smart Media cards, and Kodak alone use three different serial protocols. Their DC220, 260, 265, 290 and DC4800 all use Digita drivers, based on a different and incompatible protocol. The new and heavily-advertised Kodak DX models are based on another protocol, PTP which is public but currently only defined for USB.

The serial port interface is directly supported by my Digicam driver. This gives the greatest control, as it can issue commands to read and adjust internal camera settings, but it's not very fast - even the smallest images take a few seconds to download, and large ones may need a minute or more.

Compact Flash and IDE

But if all you want to do is read pictures from the camera into your expanded QL, Compact Flash is a good standard to look for, as it bypasses the camera's communication protocol. Cheap adapters like those offered to QL Today readers by Phoebus Dokos (QLT Volume 6 issue 5, page 8) connect the Compact Flash cards uses as removable storage by many cameras, not just those from Kodak, to an IDE interface.

You can then read the images on the memory card as if they were files on an MSDOS formatted hard disks, using Qubide, Q40, Q60, or an Amiga 600, 1200 or 4000, all of which run Qdos Classic and use the same Qdos hard disk format as

Qubide. These JPEG files can be loaded directly into Photon - all you need is a program like Disc-Over to read the MSDOS directory from Qdos, or MTOOLS or CrossDOS which comes free with Linux and AmigaOS respectively. Many cameras support Compact Flash storage - not just ones from Kodak, but Agfa, Fuji, Polaroid and others, too.

Compact Flash memory can be directly accessed at the speed of your IDE interface, but you have to swap the cards between systems and can't tell the camera to take pictures for you, or adjust settings from BASIC.

Qdos Software

DIGICAM_BAS is a 13K SuperBASIC program. The latest version is 1.6 and incorporates an improvement designed by Phil Stokes which means that even if the Qdos system is not able to keep up with the serial link all the time it can recover and re-request blocks of data if it finds that some are partly lost or corrupted. This makes it much more reliable on a slow or busy system.

Even so, you need a fast serial port, capable of reading a block of just over 1K of data without a pause at least some of the time. The standard Sinclair one can never manage this - periodically it asks the sender to stop while it transfers data elsewhere, and as the camera uses only three wires - one to send, one to receive, and one for earth - with no 'handshake' lines to pause communication, it is sure to over-run.

This photograph of Roy Wood and Jochen Merz was taken by typing the command SNAPSHOT into SBASIC on Jochen's Atari TT, after a few tweaks to the DIGICAM driver program to suit the Atari's fast SER4 port.

However the driver does work with most expanded Qdos-compatible systems, including Ataris, Amigas, Q40s and Q60s with Qdos and SMS, and a standard QL boosted with SuperHermes. It should also work with emulators like UQLX and QPC, as long as the underlying OS and serial port hardware does not get in the way.

The speed of the connection and hence the rate at which pictures are downloaded depends on the speed of the computer. Even a basic 68008 QL with SuperHermes can connect, but only at 9600 baud, around 1K per second - even so, this will load most pictures faster than they can be decoded by Photon on an eight bit QL. A Gold Card or Super Gold Card is a more practical proposition, loading at over 5K per second via SuperHermes. This is quite a usable system.

A Q40 or Q60 can manage 115,200 baud, the top speed for the camera, shifting over 10K per second. Ataris and Amigas manage rates from 19200 baud for a basic 68000 system, up to 57600 baud for accelerated 32 bit Amigas and Atari TT's. The driver was originally tested with Qdos Classic on several models of Amiga, as the interface and interrupt handler are fast enough to keep up with the uninterrupted flood of data from the camera, and then adapted for the QL when Tony Firshman kindly provided a SuperHermes. The Atari tests were performed on Jochen's machine at Quanta's AGM in Portishead, and documented in the June 2001 issue of the Quanta magazine.

Since then I've tried it on Richard Zidlicky's UQLX emulator for Linux and Unix systems, on cheap IBM clone hardware, and I'm glad to be able to report that it works perfectly from UQLX. The only changes I had to make were to specify "ser1ir" (to be on the safe side) as the serial device, and a slight tweak to the REBAUD procedure, similar to that needed for Super-Hermes, to embed the new baud rate in the OPEN string when stepping up to 115200 baud or other rates.

It works flawlessly at that speed on my Linux box, a K6/2-500 with Aladin AGPset motherboard, and I expect it will work well on anything with a buffered serial port under Linux, e.g. a 16550 or better, standard on PCs since they started using modems. The tweak gets around the upper limit of 19200 bits per second on the standard Sinclair BAUD command.
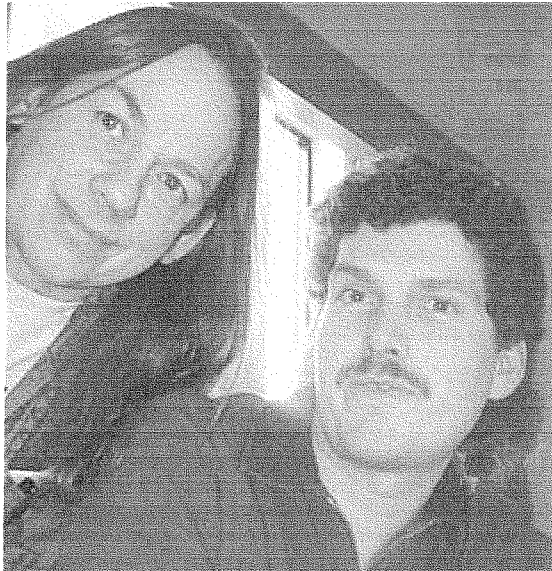
Functions

A few Turbo Toolkit commands and two new SuperBASIC extensions are used in DIGICAM_BAS; assembler source is available for both, if you're interested, and none of them are essential - they just shorten and speed up the program a bit. CHEXOR% is a function that scans a string and combines all the bytes in it, using an exclusive-or function. This is the recommended method to check that data from the camera has arrived without corruption - the result is always the same if all the bits have arrived unscathed.

Originally this was done in SuperBASIC, which was quite bearable on my 68060 Amigas, but the overhead of slicing out each character and doing arithmetic on its code was substantial, even on a 68040, and on my Gold Card I found the checksum was taking more time than the data transfer, till I recoded it in assembler. Claus Graf found that his Q40 linked to a DC215 at 115200 baud was 1.4 times faster with CHEXOR% - taking 24 seconds instead of 33 to read a 230 KB JPEG image. The whole extension is just 90 bytes long, including name table and name list initialisation.

The other new function is a variant of INPUT$ and INBYTES familiar from Turbo Toolkit, DIY Toolkit, and other extension packages. Like those it performs the equivalent of multiple INKEY$ calls from a channel, reading a specified number of bytes with a single call. Like INKEY$, but unlike the other toolkit commands, INBYTE$ supports a timeout parameter so it can read lots of bytes at once but if the data does not arrive after a specified amount of time it returns with what it can get, rather than waiting indefinitely.

This little tweak has taken several tries to get right, with help from George Gwilt, and makes DIGICAM_BAS much more resilient - Phil Stokes has also helped by working out how to get the driver to recover automatically if it gets out of sync, so it works even if badly-behaved programs or device drivers periodically block access to the serial port, reloading the 1K block till it arrives complete and with the right checksum.

Before you type RUN you need to check that the camera is switched on, and set to CONNECT, and that the wire from the camera to the chosen serial port is connected at both ends. The standard QL serial port is not capable of coping with the large blocks of data from the camera, but SuperHermes fixes that, and most later computers can cope as long as their operating system doesn't interrupt the data flow.

You do have to make sure that flow-control handshaking and translations are turned off, and may need to add extra parameters to the device name to select speeds beyond the limit of Sinclair's BAUD command, depending on the degree of QL-compatibility of your system. The code to set the port and speed is isolated in the program to make it easy to change, and even if your platform is not yet tried you should only need to add or change a few lines to make it work, hardware permitting.

Commands

When you run the program it opens the serial port on channel #3 and uses the STATUS
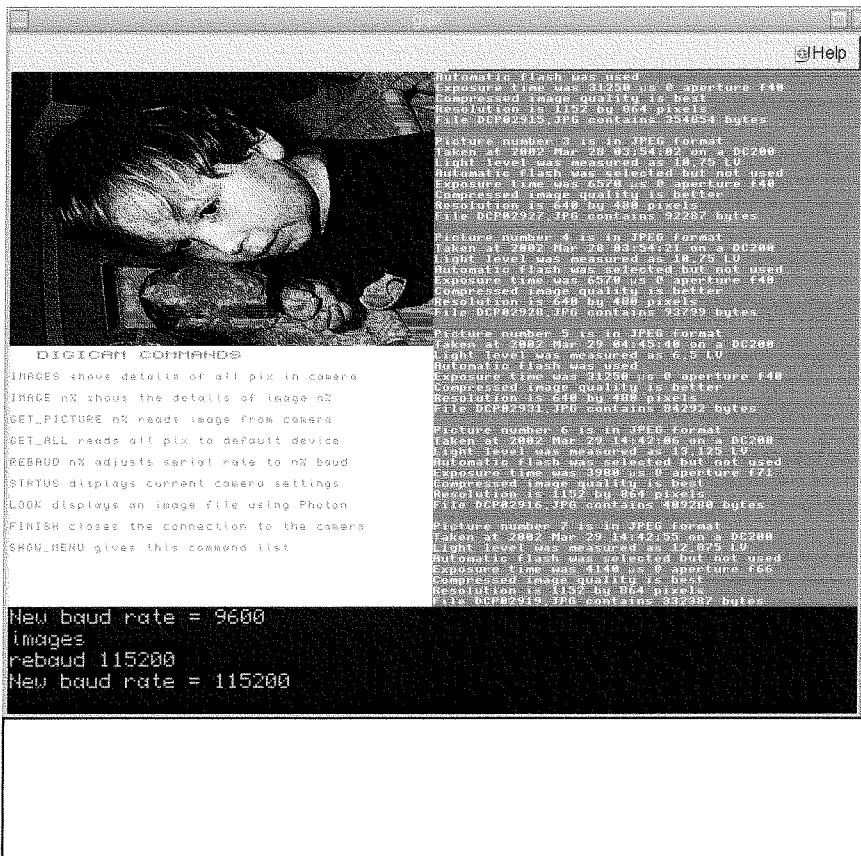
PROCedure to tell you how many pictures are in the camera, and other information. After that you can issue commands like IMAGE, GET_PICTURE, REBAUD and SNAPSHOT to control the camera.

The very simple PROCedure SNAPSHOT will take a picture, and display dots while it awaits compression to be completed. This calls another procedure SEND1 to send a command, code 124, to the camera, and then waits for a status byte from the camera. This can have one of four values - ACK once the command has been accepted, BUSY while the image is being compressed, then DONE when the command is complete or SORRY if the camera was unable to store the picture - probably because it ran out of space.

You can use the STATUS command to check that there's room before taking a picture, and IMAGE to find out the size of a given JPEG inside the camera and the settings when it was taken. At the moment this uses an index number, which changes when pictures are deleted, but if we add support for the cameras that only use filenames (like the DC240, 280 and 5000) you'll have to specify a name like DCP02912.JPG, which won't change once the photo has been stored. George Gwilt is investigating these tweaks at the moment, but help would be very welcome.

IMAGES gives details of all the pictures in the camera's memory, in a scrolling list. You can redirect this to a file with the DIY Toolkit USE command, which is how I grabbed some of the examples for this article.

GET_PICTURE reads an image from camera, given the index number (currently) or potentially a synthetic filename, while GET_ALL is a bulk downloader that reads all the pictures to the default device (e.g. WIN1_) while you get on with something else.

REBAUD n% adjusts the serial port speed to n% baud - the cameras support rates from 9,600 up to 115,200, and the faster you can use, the better. Some experimentation may be needed to find the best rate for your system. If DIGICAM_BAS reports errors at 115,200 baud try lower rates till it works reliably.

You may need to unplug the camera end of the cable, switch it off and switch on again if it get stuck in the middle of a data transfer. After this DIGICAM_BAS should resynchronise automatically when you re-run the program. Initial communi-

cation is at 9,600 baud, but that's rather slow for image downloading.

STATUS displays the current camera settings. Most of these can be changed with messages from the QL - you can even reset the Camera's clock, or (perhaps more usefully) set your QL clock from the Kodak one. The key routines to do this are in the SuperBASIC but you may need to add lines to call them with appropriate 'operation codes', depending on what you want to tweak.

The accompanying table 2 shows the output from STATUS. If you type CONTINUE after getting STATUS you'll see the details of all the images on the current card - an example is in table 1.

Table 1 - Results from the IMAGE 3 command:

Picture number 3 is in JPEG format
Taken at 1999 Dec 31 23:53:58 on a DC200
Light level was measured as 6.5 LV
Flash was selected but not used
Exposure time was 31250 microseconds · aperture f40
Compressed image quality is better
Resolution is 640 by 480 pixels
File DCP02912.JPG contains 73376 bytes

This table shows details stored with a single image in the cameras memory, as printed by SuperBASIC after an IMAGE command.

Table 2 - Results from the STATUS command:

Camera time 2002 Apr 03 07:46:52
Flash charged
Image quality is better
Resolution 640 by 480 pixels
626 pictures taken
359 flashes fired
93 pictures stored in camera
Good quality free capacity 5
Better quality free capacity 3
Best quality free capacity 1
Battery status medium
File type for new images is JPEG
Camera model is DC200
Video format is PAL
Flash mode is automatic
Exposure compensation is automatic
Automatic delay timer is off

This table shows the current settings of the camera, as discovered by typing the STATUS command in SuperBASIC.

Listing - The CHEXOR extension

```
* QDOS DIGICAM PROJECT - CHECKSUM EXTENSION CHEXOR
* Version 1.0, Copyright May 2001 Simon N Goodwin.
* Function returns byte result of exclusive or-ing
* all the character codes of a string together - a
* Kodak digital camera driver optimisation.

start        lea        define,a1
             move.w     $110.w,a2      BP.INIT vector
             jmp        (a2)

* result% = CHEXOR% ( name$ )

chexor       movea.w    $116.w,a2      Vector to get a string
             jsr        (a2)
             bne.s      bad_exit
             subq.w     #1,d3          One's enough, eh, Ingo?
             bne.s      bad_param
             move.w     0(a1,a6.1),d1  String length
             ble.s      bad_param
             lea.l      0(a1,d1.w),a0  Work out result offset
             btst       #0,d1          Is length odd?
             beq.s      scan
             addq.l     #1,a0          Round up for odd byte

* Accumulate checksum in D3 as it's conveniently empty.
* Note that EOR does not support the memory-to-register
* addressing mode, doe to the way Motorola shoehorned
* it in to the 68K instruction set. Unlike AND and OR,
* it has to share its opcode prefix group...

scan         move.b     2(a1,a6.1),d0  Fetch the datum
             eor.b      d0,d3          Mash it in directly
             addq.l     #1,a1
             subq.l     #1,d1
             bne.s      scan
return_d0    move.l     a0,$58(a6)     Update BV.RIP
             movea.l    a0,a1          Placate Turbo
             move.w     d3,0(a6,a1.1)  Stack the result
             moveq      #3,d4          Type = 16 bit Integer
             moveq      #0,d0          No run-time error
             rts

bad_param    moveq      #-15,d0        BAD PARAMETER error
bad_exit     rts

define       dc.w       0,0,1          No procedures, 1 FN
             dc.w       chexor-*
             dc.b       7,'CHEXOR%'    "string"
             dc.w       0

             end
```

The CHEXOR% extension assembles to just 90 bytes of code that make digital camera input checking a lot faster.

Other commands include FINISH, which closes the connection to the camera so you can use the port for something else, SHOW_MENU which displays a brief summary of commands, and LOOK which displays an image file, using the Photon extension PHOT.

By default this uses your current screen resolution and colours, but you can easily select another MODE in the procedure or add parameters to change the way Photon handles colours, adjusting the dithering or forcing a grey scale. This gives good results, better than newspaper quality, with a mono monitor. If you have a colour screen but want extra grey scales the PROCedure AMIGREY reprograms the Amiga palette accordingly - something similar should be possible on other emulators.

Conclusion

The driver does everything I want now, though as all the program source is freely available I hope that others will adapt and extend it to meet their own special needs. Several of the people who have helped me test it - and in some cases gone out and bought cameras just to do so! - are accomplished QL programmers, but as the driver is almost entirely written in BASIC, and details of the protocol are freely available, you don't need to be a software expert to customise the program.

Please let me know how you get on - but don't ask me for recommendations about cameras not mentioned, because I've explained what I

know - and if manufacturers don't give out details to make their cameras usable, we should not be supporting them.

All the photos accompanying this feature were originally captured on the DC200 and downloaded to Qdos Classic. After cropping to remove irrelevant borders and backgrounds, the full-colour JPEG files delivered to QL Today for printing ended up the following sizes:

CompactFlash card 4,698 bytes
DC200+ self portrait 7,673 bytes
Roy and Jochen 19,762 bytes

Table 3 - Computers, systems, speed limits and serial port setups				
Computer	CPU bits	Top speed	Serial port	QDOS
Sinclair QL	8	9,600	ser3	Sinclair/
+SuperHermes	16/32	57,600	ser3	Minerva
Amiga 500/600	16	19,200	ser1	Classic
Amiga 1200/4000	32	57,600	ser1	Classic
Atari ST	16	19,200	ser1	SMS
Atari MegaST	16	38,400	ser2	SMS
Atari TT	32	57,600	ser4	SMS
Q40/Q60	32	115,200	ser1/2	Classic/SMS
Linux	16-64	115,200	ser1/2/3/4	Sinclair
QPC	32	?	?	SMS

This table shows the ports and perfomance of systems running DIGICAM_BAS Results for QPC, Thor, QXL and Mac are solicited.

The Serial QL...

by Dilwyn Jones and Tony Firshman

Q. How do I wire up a cable to connect QL and PC via the serial ports to do file transfers?

A. Thanks to Tony Firshman (from whom you can purchase ready made cables) for the information to answer this question.

The tables below show the pin numbers for the British QL 6-pin serial ports and PC COM ports with 9 pin and 25 pin D connectors.

According to the QL manual, SER1 is configured as a DCE or Data Communications Equipment. In essence, SER1 is wired as though it was a modem. SER2 meanwhile is wired as a DTE or Data Terminal Equipment, in essence it's wired as a computer serial socket! The inputs and outputs are if effect swapped over despite the rather confusing labelling in the Communications: RS-232-C section of the Concepts part of the QL User Guide. In terms of

connecting computer to computer, since SER1 is effectively wired as a modem, it may be easier to visualise using SER2 instead.

Table 2 shows the QL serial ports pinout. Does anyone have this information for a QL such as a Samsung unit with German or American 9 pin QL connections which we can add to this article?

If you think of QL to 25D by pin number, then ser1 is "straight through" (RxD to RxD, TxD to TxD and so on, and ser2 is "crossover" (RxD to TxD, TxD to RxD and so on). That is always my starting point, and all the rest can be derived from that.

The P-Word...

To connect two PCs with COM ports running QL emulators for example (e.g. to use SERNET) you will need a crossover cable with handshake wires connected as shown in the diagram below. This may also work on QL-compatibles with serial ports wired the same as

Table 1

```
QL              QL              QL CABLE        PC          PC    PC
ser1            ser2            COLOUR          SIGNAL      25D   9D
=================================================================
3 RxD           2  TxD          white    ->     RxD          3    2
2 TxD           3  RxD          green    <-     TxD          2    3
5 CTS           4  (DTR=RTS)    blue     ->     CTS          5    8
4 (DTR=RTS)     5  (CTS)        red      <-     RTS          4    7
1               1               black    --     GND          7    5
N/A             N/A                             DSR          6    6
N/A             N/A                             DTR         20    4
```

N/A means Not Applicable, the QL does not have DSR and DTR. The so-called DTR on a QL SER1 and SER2 pin 4 is actually more of an RTS as shown above by DTR=RTS. It certainly makes wiring easier if you think of QL DTR as being an RTS.
-> and <- denote signal directions.

Table 2

SER1 (DCE) pin	name	function	SER2 (DTE) pin	name	function
1	GND	signal ground	1	GND	signal ground
2	TxD	input	2	TxD	output
3	RxD	output	3	RxD	input
4	DTR=RTS	ready input	4	DTR=RTS	ready output
5	CTS	ready output	5	CTS	ready input
6		+12 volts	6		+12 volts

GND=Signal ground (common)
TxD=Transmit Data
RxD=Receive Data
RTS=Request To Send
CTS=Clear To Send
DTR=Data Terminal Ready
DSR=Data Send Ready

The whole diagram above is QL ‹–› PC only of course.

the PC COM port equivalents, e.g. some Auroras were supplied with PC style serial connection leads. The following lines should be connected where they exist:

RxD–TxD
TxD–RxD
CTS–RTS
RTS–CTS
DTR–DSR
DSR–DTR
GND–GND

On computers which don't have DTR and DSR, leave them unconnected, in most cases it will work fine without them.

If you are experiencing difficulty getting transfer to work reliably, start with a low baudrate. If short transmissions at low baud rates seem to work, this can indicate a handshaking issue. Try to establish the correct wiring for proper serial handshaking and (software permitting) you should be able to establish successful connections.

Table 3

9 PIN COM	25 PIN COM	SIGNAL	9 PIN COM	25 PIN COM
2	3	RxD–TxD	3	2
3	2	TxD–RxD	2	3
8	5	CTS–RTS	7	4
7	4	RTS–CTS	8	5
5	7	GND–GND	5	7
6	6	DSR–DTR	4	20
4	20	DTR–DSR	6	6

Sernet Users

The above cabling should work for a two computer sernet link system. With more than two computers, you need (a) cabling which operates as a loop – third computer loops back to first for 3 computers for example and (b) a suitable sernet extensions file (some versions of Sernet only allow use of a two computer system!).

Sernet requires split input and output serial port drivers, the so-called SRX and STX receive and transmit drivers. These are normally only available in

SMSQ/E, although by using Hans–Peter Recktenwald's SimSer extensions (available for download from **http://www.soft.net.uk/dj/software/other/other.html** for example) such serial port drivers can be implemented from QDOS too (thanks to Derek Stewart for that tip).

For more details of how to use Sernet, refer back to Volume 5 Issue 1 page 32 of QL Today. Sernet software is supplied by Jochen Merz Software and can be found on most SMSQ/E master disks (ATARI, GoldCard/ SuperGoldCard, QXL and QPC 1 and 2).

A final hint from the publisher: I have lots of serial cables, connectors and adaptors to be able to connect every serial device with another serial device. Several were bought ready-made, several I soldered myself. I have always found that a serial tester with dual-colour LEDs is more than helpful. Plug it between a working connection, open the port, send some data, close it and see how the LEDs change. Once you have an idea what the LEDs should look like, you will find it easy to make up own cables or set up new connections. Serial testers don't cost much, and they can and will save you so much time!

Hints and Tips

1. Larger Editing Windows

Q. Is it possible to increase the area of the window used for editing BASIC programs? I dislike having to edit programs in a small box in the top left of a high resolution screen and would like it if there was a command to enable BASIC programs to be edited using the entire screen.

A. Yes, the LIST, EDIT and ED keywords can all cope with large windows. Since you are using high resolution screens, I'll assume you are using SMSQ/E. If you were happy with the size of windows but wanted them away from the top left of the screen, it is possible for the WMON and WTV extensions to have offsets, to move the windows away from their normal position. For example,

```
WMON 4,50,50
```

will act like a normal WMON 4 command, but the windows will be at a position determined by the two numbers after that, in case 50,50. Of course it is easy to issue three WINDOW commands to enlarge the windows, but this is tedious and you have to remember the different sizes for each window in each screen resolution.

Back in the mists of time (well, QL Today Volume 2 Issue 3 actually) I wrote a set of extensions called DISPLAY_CODE which returned the maximum sizes of a window. FLIM_W(#0) and FLIM_H(#0) are functions which tell you the largest size the #0 window could take (the full screen size). SBASIC also has the equivalent SCR_XLIM and SCR_YLIM functions.

It is then a simple matter to write a small BASIC procedure you can add to the end of your BASIC programs to calculate the window sizes needed. If you do not have the FLIM_W and FLIM_H extensions, simply use SCR_XLIM and SCR_YLIM instead – I used FLIM_W and FLIM_H to enable the procedure to work on QDOS systems with pointer environment as well. OUTLN is an extension for setting the saved outline of windows, some systems may not need this.

The OUTLN extension is present in SBASIC and also supplied in many toolkits and pointer driven software packages – QMenu has an OUTLN_rext extension for example. If it gives a 'bad name' or similar error, try removing line 1010 to see if it will work without it.

```
1000 DEFine PROCedure XTV
1010   OUTLN #0,FLIM_W(#0),FLIM_H(#0),0,0
1020   WINDOW FLIM_W(#0),FLIM_H(#0)-52,0,0
1030   WINDOW
#2,FLIM_W(#0),FLIM_H(#0)-52,0,0
1040   WINDOW #0,FLIM_W(#0),52,0,FLIM_H(#0)
       -52
1050   BORDER #1,1,255 : BORDER #2,1,255 :
       BORDER #0,1,255
1060   CLS : CLS #2 : CLS #0
1070 END DEFine XTV
```

Because I normally use the TV mode windows for BASIC program editing, I've based this routine on the assumption that #1 and #2 will use the same area of screen. if you prefer monitor mode windows, modify line 1020 to set #1 to use half the screen width, halfway across the screen and 1030 to use half the screen width on the left of the screen:

```
1020   WINDOW FLIM_W(#0)/2,FLIM_H(#0)-52,
       FLIM_W(#0)/2,0
1030   WINDOW #2,FLIM_W(#0)/2,FLIM_H(#0)
       -52,0,0
```

2. Type in Text

Q. I would like to send some text from a program I have written directly into another program as though it had been typed into that program – how can I do this?

A. Several toolkits have their own extensions such as TYPE_IN to do this. However, if you do not have one of these extensions, or it does not work on your system (many such extensions do not work properly on pointer environment systems due to the way PE manages keyboard queues etc), it is possible to use the stuffer buffer of the hotkey system to transfer small amounts of text.

The QPAC2 manual has a wonderful example, a short two line compiled BASIC program which can insert the date string into any program on page EE44 of the manual, based on a short program like this:

```
100 HOT_STUFF "string"
110 HOT_DO " " : REMark one space inside
the    quotes
```

Line 100 inserts the string into the stuffer buffer, then line 110 pretends that ALT SPACE has been used to extract the string from the stuffer buffer. The keyboard queue needs to be switched to the

DILWYN JONES

QL P.D. SOFTWARE LIBRARY SERVICE

THE TOP 10 PACKAGES!

1. GE01 - Turbo Compiler
2. LI02 – CIA World Factbook (5 disks)
3. GE51 – The Editor 2000 (2 disks)
4. GE53 – Atapi/CDROM package for Qubide, Q40 to access CDs (2 disks)
5. GA01 – Solvit Plus word games (2 disks)
6. GE25 – DIY Toolkit (3 disks)
7. GE13/14 – QL Info-Unzip/Zip (1 disk each)
8. GE32 – C68 Compiler (6 disks)
9. GE03 – TurboPTR (2 disks)
10. GA05 – Radar, Flightdeck demo and QL War game packages

The QL PD Library consists of:

- Over 60 disks assorted PD programs
- 15 disks with over 100 games in all!
- QL manuals and information on disk
- QL Hackers Journal & ClubQL e-zines
- 46 disks of QL screens clipart
- 26 demos of commercial QL programs
- Back issues of QL Today cover disks
- QL emulators for several computers
- Loads of Line Design clipart and fonts
- Loads of classic literature on disk!
- Lots more than I can list here!
- Just £1 per disk (or 75p if you supply disk) plus postage – see below
- Catalogue available – see below

For a PD software library catalogue, send me a formatted floppy disk, or an email requesting it to be sent to you by email, or download a catalogue from my website (address below). This PD library is run as a non-profit making service for the QL community. All programs in the PD library are PD, Freeware, or Shareware etc, also freely available from other libraries and websites, some also available for free download on my website where space permits.

QL CD-ROMS

NEW QL DOCUMENTATION CD £5.00 – Vast collection of manuals, documents, articles, beginners information etc ***NEW***
DJ PD LIBRARY CD £10.00 – My entire PD library on just one CD!
QL EMULATORS CD £5.00 – All of the available emulators on one CD-ROM!
QL PD-CDR £5.00 – Gérard Plavec's vast software collection.
LINE DESIGN CLIPART CD £10.00 – Huge collection of Line Design clipart
FAMOUS FACES CD £10.00 – Line Design pictures of the famous!
QL RELIGION CD £10.00 – A collection of Bibles, religious texts, cliparts etc
CD-ROMs supplied in ISO-9660/Joliet format (Rockridge by special request) for use with most systems able to access QXL.WIN media. Q40/Qubide 2 users may be able to use Thierry Godefroy's Atapi/CD drivers (disks GE53 in the library) to access these CDs.

POSTAGE UK/Europe: add £1.00, Rest Of World: add £2.00, per order.
PAYMENT In UK Pounds Sterling currency only please to avoid bank charges for currency conversion. Cheques payable to DILWYN JONES.

**Dilwyn Jones, 41 Bro Emrys,Tal-y-bont,Bangor,Gwynedd, LL57 3YT, U.K.
Email:dilwyn.jones@dj.softnet.co.uk
Website: http://www.soft.net.uk/dj/index.html**

Note: although run by the editor of QL Today,the PD library is a separate venture with no connection to the publisher of this magazine. Software is supplied free of charge, apart from copying and media charges etc to cover costs and without any warranty other than to the replacement of defective media. In other words, usual PD library terms. E & OE

right program, so your program needs to be either unlocked or have no open windows at the point this code is executed.

Directory Deletion
- update from David Denham
Some time ago, I gave a little hint on deleting a single file or directory name in QPAC2, based on selecting multiple files and selectively deleting them to get around the fact that if you DO on a single directory name, QPAC2 tries to enter that directory rather than bringing up the commands menu to let you get at the DELETE command. In fact, there is a simpler solution. HIT the directory name, then DO on a part of the files window which isn't a filename, in other words, right click on nothing. The commands menu now comes up, letting you remove the directory name (as long as the directory is empty).

QLTdis - part 6 / 2
Norman Dunbar

The next routine is the start of the effective address decoding stuff. I though that this would give me the most grief, but as it turned out, it was actually quite simple to do. Remember an effective address is part of the op-code and uses bits 0 to 5 to hold a register number and a mode number. In this routine we simply extract these two from the op-code into D0.W (the mode) and D1.W (the register) and then use yet another jump table to fire us off into another part of the code.

```
*————————————————————————————————————————————————————————————————————*——
* Effective Address decoding subroutine
*————————————————————————————————————————————————————————————————————*——
eff_addr     move.w  d0,-(a7)          ; Preserve D0.W
             tst.w   d5                ; Has the size been set yet ?
             bne.s   ea_sized          ; Yes – already set
             move.w  d0,d5             ; Save D0 for now
             bsr     size_d0           ; Get size in D0
             exg     d0,d5             ; Get size in D5 and restore D0

ea_sized     clr.l   d1                ; D1 will be the register eventually
             move.w  d0,d1             ; D0 is the mode
             andi.w  #$38,d0           ; Keep only the mode bits in 3 - 5
             lsr.w   #2,d0             ; D0 = mode * 2 = offset into mode_table
             andi.w  #$07,d1           ; Keep only the register bits
             lea     mode_table,a3     ; Mode table address in A3
             move.w  0(a3,d0.w),d2     ; Fetch the offset to the mode routine
             adda.w  d2,a3             ; A3 now points at the mode routine
             jsr     (a3)              ; Process the mode
             move.w  (a7)+,d0          ; Restore D0
             rts

mode_table   dc.w    mode_0–mode_table
             dc.w    mode_1–mode_table
             dc.w    mode_2–mode_table
             dc.w    mode_3–mode_table
             dc.w    xmode_4–mode_table
             dc.w    mode_5–mode_table
             dc.w    mode_6–mode_table
             dc.w    mode_7–mode_table
```

Here is the mode 0 routine – it deals with Data Register Direct addresses and simple adds 'Dn' to the buffer. Start simple and work up !

```
*————————————————————————————————————————————————————————————————————*——
* Mode 0 – Data register direct
*————————————————————————————————————————————————————————————————————*——
mode_0       bsr     dddd              ; Its data register direct
             move.w  d1,d4             ; Data register number
             bsr     reg_no            ; Must be 0 to 7
             rts                       ; Done
```

Another simple one for mode 1 – its Address Register Direct this time which simply adds 'An' to the buffer.

```
*─────────────────────────────────────────────────────────────────
* Mode 1 - Address register direct
*─────────────────────────────────────────────────────────────────
mode_1      bsr     aaaa        ; Its Address register direct
            move.w  d1,d4       ; Address register number
            bsr     reg_no      ; Must be 0 to 7
            rts                 ; Done
```

Starting to get harder not, mode 2 is the Address Register Indirect mode, and adds '(An)' to the buffer.

```
*─────────────────────────────────────────────────────────────────
* Mode 2 - Address register indirect
*─────────────────────────────────────────────────────────────────
mode_2      bsr     l_bracket_a ; Add '(A' to the buffer
            move.w  d1,d4       ; regsiter number
            bsr     reg_no      ; Add to the buffer
            bsr     r_bracket   ; Close bracket
            rts                 ; Done
```

Adding to the complexity, we have mode 3 – Address Register Indirect with Post-Increment, which is basically, a mode 2 with a plus sign on the end – so guess how we decode this then? Correct – we call the mode 2 routine and add a '+' to the buffer. Simple and effective. (Opps, sorry that pun was intended!)

```
*─────────────────────────────────────────────────────────────────
* Mode 3 - Address register indirect with post-increment
*─────────────────────────────────────────────────────────────────
mode_3      bsr     mode_2      ; Pretend address register indirect
            moveq   #'+',d4     ; And now the post-increment bit
            bsr     str_add_b   ; Add to the buffer
            rts                 ; Done
```

Mode 4 gives us another chance to re-use some code. Address register Indirect with Pre-Decrement is a minus sign followed by a mode 2, so thats how we decode it.

```
*─────────────────────────────────────────────────────────────────
* Mode 4 - Address register indirect with pre-decrement
*─────────────────────────────────────────────────────────────────
xmode_4     moveq   #'-',d4     ; First the pre-decrement bit
            bsr     str_add_b   ; Add to the buffer
            bsr     mode_2      ; Pretend address register indirect
            rts                 ; Done
```

I'm beginning to see a pattern developing here, mode 5 is ARI (I can't be bothered typing it all out again!) with a number in front. The number is a 16 bit displacement and has to be extracted from the word that follows the op-code. Then, we call our mode 2 routine again to get something like 'dddd(An)' in the buffer.

```
*─────────────────────────────────────────────────────────────────
* Mode 5 - Address register indirect with displacement
*─────────────────────────────────────────────────────────────────
mode_5      bsr     dollar      ; Hex addresses need a dollar
            move.w  (a6)+,d4    ; Fetch the displacement
            bsr     d4_hex_w    ; Add to the buffer
            bra.s   mode_2      ; Pretend address indirect mode
```

Ah well, it had to end sometime. No more mode 2 stuff. Now we are into mode 6 which is Address Register Indirect with Index. This is something like mode 5 above but the displacement is only 8 bits, but we have an index register – which can be word or long sized to consider as well. This adds something like 'dd(An,Rx.size)' to the buffer.

```
*———————————————————————————————————————————————————————————————
* Mode 6 - Address register indirect with index
*———————————————————————————————————————————————————————————————
mode_6      bsr     dollar          ; Hex addresses need a dollar
            move.w  (a6)+,d0        ; Get the index & displacement data
            move.w  d0,d4
            bsr     d4_hex_b        ; Add the displacement byte
            bsr     l_bracket_a     ; Add '(A' to the buffer
            move.b  d1,d4           ; Get the register number
            bsr     reg_no          ; Add to the buffer
            bsr     comma           ; Followed by a comma
            move.w  d0,d4           ; Get the index data again
            bsr     index_reg       ; Add to the buffer
            rts                     ; Done
```

Mode 7 is not a real mode, but a catch all for stuff not covered above. The register number we so carefully stored in D1.W is not really a register number, but a sub-mode. The problem is that while there are 8 values possible in D1, only 5 are valid. If the register number is 5, 6 or 7 then it is an invalid mode.

Once again, we simply use D1.W as an offset into a table and jump into hyperspace again.

```
*———————————————————————————————————————————————————————————————
* Mode 7 - Select the sub-mode routines
*———————————————————————————————————————————————————————————————
mode_7      lea     sub_modes,a3    ; Sub-mode table
            move.w  d1,d2           ; Get the sub-mode number into D2
            lsl.w   #1,d2           ; Offset into table
            move.w  0(a3,d2.w),d2   ; Fetch the offset
            adda.w  d2,a3           ; Calculate the routine address
            jsr     (a3)            ; Call the sub-mode address
            rts                     ; Done

sub_modes   dc.w    sub_mode0-sub_modes
            dc.w    sub_mode1-sub_modes
            dc.w    sub_mode2-sub_modes
            dc.w    sub_mode3-sub_modes
            dc.w    sub_mode4-sub_modes
            dc.w    sub_mode5-sub_modes
            dc.w    sub_mode6-sub_modes
            dc.w    sub_mode7-sub_modes
```

Sub-mode 0 is the absolute short addressing mode which simply adds '$aaaa' to the buffer. The word for the address is extracted from the word following the op-code.

```
*———————————————————————————————————————————————————————————————
* Sub Mode 0 - Absolute short address
*———————————————————————————————————————————————————————————————
sub_mode0   bsr     dollar          ; Dollar for addresses
            move.w  (a6)+,d4        ; Get the short address
            bsr     d4_hex_w        ; Add to the buffer
            rts                     ; Done
```

Sub-mode 1 is the absolute long address mode which adds '$aaaaaaaa' to the buffer. The long word for the address is extracted from the long word following the op-code.

```
*———————————————————————————————————————————————————————————————
* Sub Mode 1 - Absolute long address
*———————————————————————————————————————————————————————————————
sub_mode1   bsr     dollar          ; Dollar for addresses
            move.l  (a6)+,d4        ; Get the long address
            bsr     d4_hex_4        ; Add to the buffer
            rts                     ; Done
```

Sub-mode 2 adds 'Laaaaaaaa(PC)' to the buffer and is of course, the PC with displacement addressing mode.

```
*————————————————————————————————————————————————————————*
* Sub Mode 2 - PC with displacement
*————————————————————————————————————————————————————————*
sub_mode2   bsr     ell             ; Add 'L' to the buffer
            move.l  a6,a3           ; Get current PC address
            adda.w  (a6)+,a3        ; Add offset word (sign extends)
            move.l  a3,d4           ; D4 has the destination address
            bsr     d4_hex_4        ; Add to the buffer
            move.l  #'(PC)',d4      ; Some text
            bsr     str_add_l       ; Add it
            rts                     ; Done
```

Sub-mode 4 is PC with index and is similar to the above, but involves an index register as well. As you may have surmised by now, the index register bit is handled by another sub-routine which will be described later. This routine adds stuff like 'Laaaaaaaa(PC,Rx.size)' to the buffer.

```
*————————————————————————————————————————————————————————*
* Sub Mode 3 - Program counter with index
*————————————————————————————————————————————————————————*
sub_mode3   bsr     ell             ; Add 'L' to the buffer
            movea.l a6,a3           ; Get current PC address
            move.w  (a6)+,d0        ; Get index & displacement data
            move.w  d0,d4
            ext.w   d4              ; Extend the byte to a word
            adda.w  d4,a3           ; Add to current PC address
            move.l  a3,d4           ; Ready for conversion
            bsr     d4_hex_4        ; Add to the buffer
            move.l  #'(PC,',d4      ; PC part of the instruction
            bsr     str_add_l       ; Add to the buffer
            move.w  d0,d4           ; Copy again
            bsr     index_reg       ; Call index register routine
            rts                     ; Done
```

Sub-mode 4 is quite simple, but needs the value in D5.W to be set to 1, 2 or 4 as per the decoding of sizes routine above. This value in D5.W tells the following code how many bytes are to be extracted from the memory address following the op-code for the data. Long words (D5=4) need 4 bytes, word need 2 and bytes need 1. The value in D5.W is the actual amount of data to be extracted.

```
*————————————————————————————————————————————————————————*
* Sub Mode 4 - Immediate data
*————————————————————————————————————————————————————————*
sub_mode4   movem.w d0/d5,-(a7)     ; Preserve D0 and D5
            bsr     hash_dollar     ; Some immediate hex data coming next
            cmpi.b  #4,d5           ; LONG sized ?
            bne.s   s4_word         ; No - try word

s4_long     move.l  (a6)+,d4        ; Get the long data
            bsr     d4_hex_4        ; Add to buffer
            bra.s   s4_done         ; Skip

s4_word     move.w  (a6)+,d4        ; Get the word or byte data
            cmpi.b  #1,d5           ; Byte sized ?
            beq.s   s4_byte         ; Yes, skip
            bsr     d4_hex_w        ; Add to the buffer
            bra.s   s4_done         ; Skip

s4_byte     bsr     d4_hex_b        ; Convert the byte data

s4_done     movem.w (a7)+,d0/d5     ; Restore working registers
            rts                     ; Done
```

The comments for the remaining sub-modes routines explain it all – what more do I need to say? They all cause an error message to be logged to the output buffer showing the fact that a mode 7 effective address was invalid.

```
*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*
* Sub Modes 5,6 & 7 - invalid mode/register data - ERROR - the register number
* is invalid for a mode 7 instruction so add 'M7,Rn ‹——DATA ???' to the
* buffer to indicate a problem.
*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*
sub_mode5   equ    *
sub_mode6   equ    *
sub_mode7   lea    output+2,a5      ; Reset output buffer
            moveq  #0,d6            ; Reset the size as well
            move.l #'M7,R',d4       ; Mode 7 Register text data
            bsr    str_add_l        ; Add to the buffer
            move.b d1,d4            ; Get the register number
            add.b  '0',d4           ; Convert to a digit
            bsr    str_add_b        ; Add to the buffer
            move.l #' ‹—',d4        ; D4 = 'space arrow'
            bsr    str_add_l        ; Add to the buffer
            move.l #'DATA',d4       ; Text
            bsr    str_add_l        ; Add to the buffer
            move.l #' ???',d4       ; More text
            bsr    str_add_l        ; Add to the buffer
            rts                     ; Done
```

This next routine is the famous index register sub-routine as used in a number of places above. It extracts the index register details – address or data – and size – word or long – from the word in memory, and adds it all to the buffer. We generate stuff like 'An.size)' or 'Dn.size)' in this routine.

```
*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*
* Index Register sub-routine
*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*
index_reg   btst   #15,d0           ; Address or data ?
            beq.s  ir_data          ; Clear = data
            bsr    aaaa             ; Do an address register
            beq.s  ir_reg           ; Done

ir_data     bsr    dddd             ; Do a data register

ir_reg      andi.w #$7800,d4        ; Keep bits 12 to 14 only
            lsr.w  #8,d4            ; Shift bit 12 -> bit 0
            lsr.w  #4,d4            ; 8 bits maximum though !
            bsr    reg_no           ; Store the register number
            btst   #11,d0           ; Check the size
            beq.s  ir_word          ; Clear = word
            move.l #'.L)',d4        ; Do long size
            bra.s  ir_size          ; Skip

ir_word     move.l #'.W)',d4        ; Do word size

ir_size     bsr    str_add_3        ; Finish the op-code
            rts                     ; Done
```

A very simple routine next, which takes the value in D4 – which will be between 0 and 7 – and converts it into the ASCII digit for that value. This is then added to the buffer.

```
*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*
* Register number sub-routine.
*––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––*
reg_no      add.b  #'0',d4          ; Convert d4 from 0 to 7 -> '0' to '7'
            bra    str_add_b        ; Add it ti the buffer
```

And finally, some small but perfectly formed routines which are called from a number of places to do a simple thing – to add one or more characters to the output buffer.

When I was writing QLTdis I found that I was writing some code more than once – and this is a good sign that a sub-routine is needed. After a time, I ended up going through all the code I had, extracting all the sub-routines and counting the number of calls to them. Anything with 2 or more calls was extracted and added below here. Anything with one call stayed where it was.

Note that rather than calling 'BSR STR_ADD_x'

and then adding an 'RTS' instruction to the following routines, I simply 'BRA STR_ADD_x' which saves me a few bytes for each routine and saves typing – when the STR_ADD_x routine gets to its own RTS, control returns to wherever it was when it called the routines below.

This is quite a useful trick, but you have to know your code inside out before you can use it – because there will always be a place where it won't work due to something being on the stack above the return address – beware!

On with the rest of the code.

```
*─────────────────────────────────────────────────────────────────────
* Some useful sub-routines to add certain character strings to the output area.
* These are here if they are needed two or more times. Names should be fairly
* meaningful !
*
* NOTE : These routines are all called by a BRS instruction. As they all end up
* calling another subroutine, the BRS to that routine has been replaced by a BRA
* which means that the return to the caller of THIS routine is from the routine
* called from here ! This saves me typing RTS for every subroutine AND saves two
* bytes and 16 clock cycles per RTS instruction AND the BRA instruction is
* quicker by 8 clock cycles too !
*─────────────────────────────────────────────────────────────────────
comma       moveq    #',',d4        ; Comma required
            bra      str_add_b      ; Add it & return

space       moveq    #' ',d4        ; Space required
            bra      str_add_b      ; Add it & return

dddd        move.b   #'D',d4        ; Data register required
            bra      str_add_b      ; Add it & return

aaaa        move.b   #'A',d4        ; Address register required
            bra      str_add_b      ; Add it & return

ell         move.b   #'L',d4        ; L size required
            bra      str_add_b      ; Add it & return

uu          move.b   #'W',d4        ; W size required
            bra      str_add_b      ; Add it & return

dollar      move.b   #'$',d4        ; Hex data required
            bra      str_add_b      ; Add it & return

slash       move.b   #'/',d4        ; Slash required
            bra      str_add_b      ; Add it & return

r_bracket   move.b   #')',d4        ; Right bracket required
            bra      str_add_b      ; Add it & return

eight       move.w   #'08',d4       ; Eight required
            bra      str_add_w      ; Add it and return

comma_d     move.w   #',D',d4       ; Comma D required
            bra      str_add_w      ; Add and return

comma_a     move.w   #',A',d4       ; Comma A required
            bra      str_add_w      ; Add and return
```

```
space_d       move.w   #' D',d4         ; Space D required
              bra      str_add_w        ; Add and return

space_a       move.w   #' A',d4         ; Space A required
              bra      str_add_w        ; Add and return

dot_w         move.w   #'.W',d4         ; Size W required
              bra      str_add_w        ; Add and return

dot_ell       move.w   #'.L',d4         ; Size L required
              bra      str_add_w        ; Add and return

l_bracket_a   move.w   #'(A',d4         ; Address reg indirect etc required
              bra      str_add_w        ; Add and return

hash_dollar   move.w   #'#$',d4         ; Immediate Hex data required
              bra      str_add_w        ; Add and return

comma_ccr     move.l   #',CCR',d4       ; Move to CCR required
              bra      str_add_l        ; Add & return

comma_mba     move.l   #',-(A',d4       ; Guess !
              bra      str_add_l        ; Add & return

mba           move.l   #'-(A',d4        ; Guess again !
              bra      str_add_3        ; Add & return

comma_sr      move.l   #',SR',d4        ; Move to SR required
              bra      str_add_3        ; Add & return

dot_s_space   move.l   #'.S ',d4        ; Short branch etc required
              bra      str_add_3        ; Add and return
```

And there you have it. Type that above code into DISS_ASM and assemble the whole lot again and hopefully you will be able to disassemble some instructions in the type 0 to type 16 range.

I have tested the routines for those and they work fine, so happy disassembling.

See you next time for more exciting stuff.

QDT: A Progress Report
Jim Hunkins

I am happy to say that, despite loosing a couple of months of potential development time recently, QDT is progressing once again and getting interesting.

Just to let everyone know where things stand now, QDT can at this moment execute programs, open and update properties with tabbed property notebooks, and do some nice little things like move icons on the desktop and rename their text. And it is evolving constantly. As I implement different pieces, I keep getting different ideas of
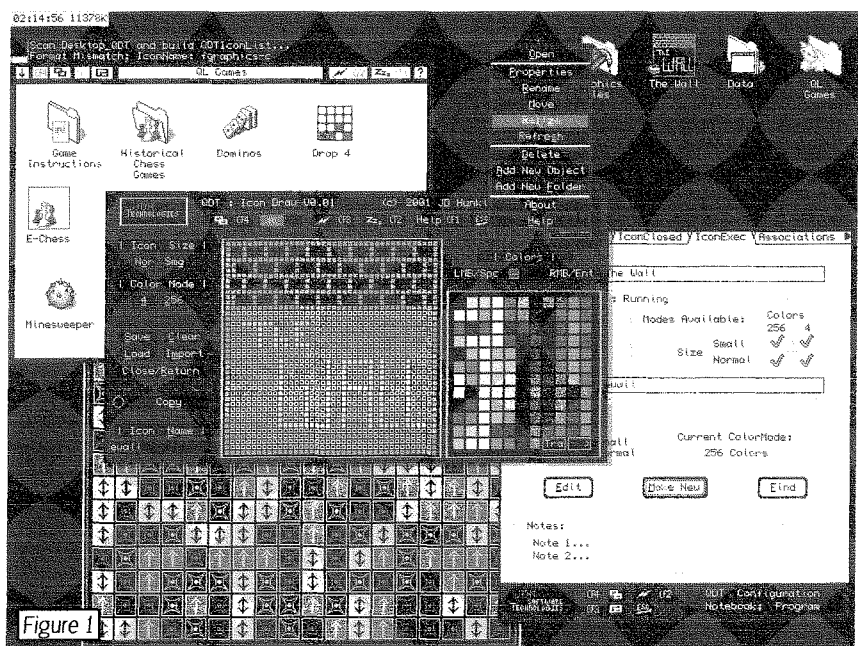

Figure 1

how to do things and what features/capabilities should be included. The hardest part is to not try to do everything up front. I am getting in the habit of taking notes for many of my ideas. But some just can't be put off. Oh, and yes, I do reserve the right to change anything you see between now and when I ship this thing. In fact, at least one of the images you will be seeing here is scheduled for modification shortly. To give everyone a snapshot of what QDT is looking like now, I am including a desktop snapshot along with several clips from different configuration notebooks. Here are some notes to help you walk through the images.

Figure 1 (previous page): QDT in Action

This is the REAL thing. QDT has opened up, setup the icons on the desktop (upper right corner) and opened a folder (upper left side). The different things that are happening here are:
1) I have started a game (lower left side) using "The Wall" icon on the desktop.
2) I have opened the properties notebook (lower right side) for "The Wall" icon and have, from the notebook page labeled "IconExec", decided to change the icon in use by pressing the "Make New" button in the property notebook page. This opened the Icon-Draw program (center). Notice how the "Make New" button is shaded? This means that the notebook is waiting until the button's action is completed; i.e. the IconDraw program closes.
3) I also pulled down the menu from the left most icon on the desktop (top center) in order to do some action with it (such as

Open the folder, rename the icon, or even delete it). Notice that the Resize option in the drop down menu is blocked out since the folder is not open and resizing doesn't make sense.
4) I have also placed my cursor back over "The Wall" icon again, but you will notice that it is locked out. While you can execute multiple programs from the same icon, once you open the property notebook for an object, you must close the property notebook before accessing that particular icon again. This makes sense because you normally would not change a property for something while trying to use it for something else (take my word for it :). While this may look complex to some of you, it is all very easy. QDT is handling all the details. Everything is just a click or two away and very simple to work with!
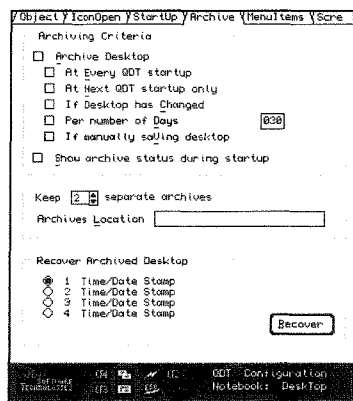

Figure 2: The QDT Desktop Archive Notebook Page

Any desktop design must have an easy way to organize your daily activities and manage the basics of computer life. It also must provide a level of security in its use and basic system management. The QDT Desktop has its own Properties Notebook which includes everything from Archiving your desktop to handling startup

options to managing screen resolution and screen savers. It will even give you system information without having to dig up a program that is buried some where on one of your disks. The archive page is a good example to look at. Using a combination of check lists, a rotating menu, and an exclusive list, you have full control over when (and if) QDT archives your current desktop setup, gives you a choice of how many archives to keep and where to put them, and allows you to recover (in case you just happen to erase one of those very important files!).
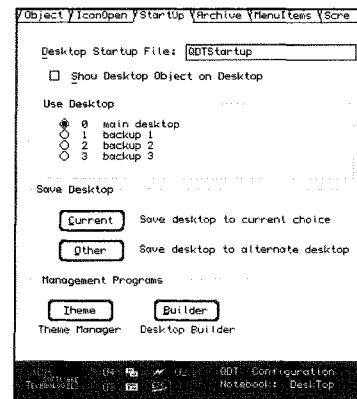
Figure 3: The QDT Desktop Startup Notebook Page

As with any program, you have a choice with QDT to configure lots of stuff (or to just leave everything as it defaults to if you wish). The Startup page gives you a quick view of what can happen when you start QDT. You can run a separate file which can run different programs automatically as part of the desktop start sequence. You can also choose whether or not to have QDT have its own icon (object) on the desktop (you will also be able to access QDT stuff from a hot key action if you don't choose this). In case you want to setup your desktop for different activities or there are multiple

users on one machine, you can have different versions of the desktop. Of course to do this, you need to be able to save your desktop. In addition, there will be automatic save options for when you change things (or you can tell QDT to forget what you just did – handy at 3 in the morning). To help you with your desktop creation, Desktop Builder (included with initial release) and Theme Manager (to be released later) can both be launched from this page.
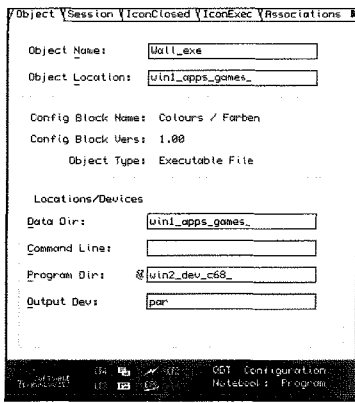
fied but there was no Command Line given (and no defaults for it). Looking at the Program Directory line, you will note a small symbol (sd with an underline) just to the left of the text box with "win2_dev_c68_". The symbol means that the notebook is using a default value since it did not find one that you gave. In this case, the notebook did not find a QDT default value but did find a system global variable value for the Program Directory. You can also prevent this default choice by simply entering an empty string into the box.
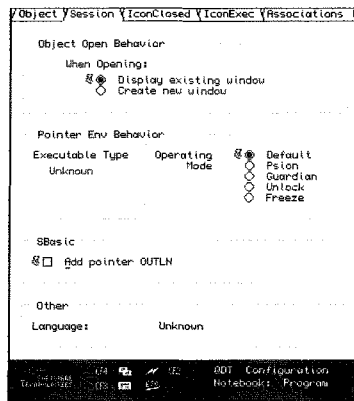
which are shown in the Pointer Env Behavior section. The notebook will try to figure out what kind of program you have, but you can always override the value by clicking on one of the exclusive list buttons on the right. You can also tell the system to setup a pointer OUTLN for an older SBASIC program. And now for the change I hinted at before. Forget the Language piece – not very useful after all so I will replace it with some other cool stuff (visit my website **www.jdh-stech.com** to find out once the change is in).

Figure 4: <u>The QDT Program Object Notebook Page</u>

This page occurs for all program, file and folder objects in one form or another. It will always give the name of the actual object and where it exists. The config block information is pulled from the object automatically if it exists (if not, there will be text telling you that no config block was found. And the object type (Executable, File (and type), Folder, etc) is given. All object types will have at least some of the Locations/Devices shown here. If nothing is specified by you (or by the automated setup during install), the notebook will check for global properties, first within QDT itself and then system wide. Here, the Data Directory was actually speci-

Figure 5: <u>The QDT Program Session Notebook Page</u>

When a program is opened, there are different ways it can be executed. This page gives you control over this without having to remember all the ways to make these happen. First, you can tell QDT to always use the same existing program if it already has been run (basically this will pick the program and put your cursor onto it). Or you can tell it to run a new copy everytime, regardless. And note the "sd" symbol next to the check box. This again means that you did not make a selection so a default value is being used. The pointer environment has different things it can do to help with different types of programs,

Figure 6: <u>The QDT Program IconClosed Notebook Page</u>

Every object needs a pretty icon, someone once said (OK, it was me a few minutes ago). And here is where you have control over what you have to look at. You can set the Icon text here, change the Icon (which is shown), and even pick a different size. The modes available in the actual Icon file are shown for you convenience. And if you feel really creative, hit the Edit or Make New button to update the Icon itself. If you aren't sure which Icon to use, hit the Find button to bring up an Icon Browser (on the list of programs for later release). The notes section will alert you if there was any problem and a

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12   D-47169 Duisburg
Tel. 0203 502011   Fax 0203 502012
http://www.j-m-s.com/smsq/index.htm

SMSQ/E Prices

Due to changes in the licensing structure we are happy to be able to reduce the prices for all SMSQ/E products!

QPC2 Version 3 with SMSQ/E

New Version with complete manual   **EUR 99,90**

Upgrade to QPC2 Version 3

From QPC1 (or just QPC - the first QPC!)   **EUR 69,90**
From QPC2 Version 1 (or just QPC2)   **EUR 44,90**
From QPC2 Version 2   **EUR 20,40**

You will receive a new QPC part of the manual. Please return QPC master disk. Please note: QPC1 will not be supported on this disk anymore. If you want to continue to use QPC1, then please make a backup-copy of the QPC master disk before you return it to us.

All other SMSQ/E

for GoldCard/SuperGoldCard or QXL or ATARI
price per Version   **EUR 49,90**

substitution was made (such as the color mode and size you requested didn't exist in the Icon File).
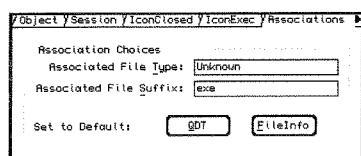

Figure 7: The QDT Program Associations Notebook Page

Many users use FileInfo currently to allow associations between types of files and executable programs (IE: click on a file of a certain type, and a specific program is started with the file chosen loaded). I am happy to say that QDT will be allowed to use the FileInfo system for its associations basis. The idea is that, for those who have FileInfo setup, QDT will inherit the settings and allow the user to configure things either with the original FileInfo or within specific notebooks. This page is just a hint of things to come and still needs to be flushed out.
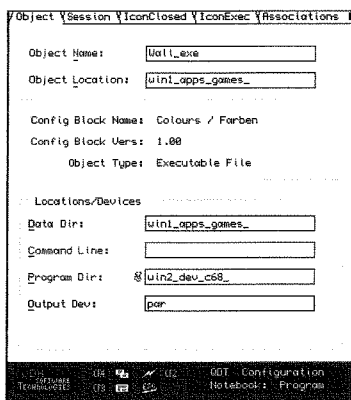

Figure 8: The QDT Folder Object Notebook Page

If you noticed, this page does not have its text boxes all filled in. All of the Program Notebook Pages shown previously were generated live by QDT. This one however is only partially implemented. What it means simply is that, while lots of progress has been made, there is still tons of work left to do. OK, enough for the caution (i.e. QDT is not shipping next week). On this page, you can see another detail of how things are being looked at. Folders in QDT can be 'normal' or 'executable'. From the notebook page, you can literally switch back and forth between

the two types by simply pressing the Change button on this page. In this case, the notebook will then switch to the other type and let you enter the executable object name, etc. Simple.

I should end this little update, as it is past 3 in the morning and I had promised it to Jochen on Sunday (yesterday technically). Since I haven't gone to bed yet, as an engineer, it is still Sunday to me. By the time you read this, I will have given a working demo of QDT as it stands on June 1 at the North American QL show. I expect my website www.jdh-stech.com will have updates about two weeks later with lots more than you see here. This will include a sample or two of some very nice cosmetic improvements to both the buttons and the actual windows in use (thanks to Marcel's recent work!). So stay tuned and feel free to drop me a note with suggestions, ideas, or whatever. And thanks to those of you who have already written in, especially those offering their assistance and responding to the questionnaire published recently.

Source Code to be available - the continuing story

Wolfgang Lenerz

I'd like to keep you informed of the state and status of the publication of the SMSQ/E source code.

The future licence-to-be has been a bit modified, notably to take into account the fact that test versions must be easily distributed. The (still provisional) text was published on Ql-users, and is as follows:

Official statement

1) This software, called 'SMSQ/E', is copyright (c) Tony TEBBY 1987-2002. Any unauthorized

copying or use of the software, whether in binary or source code form, and/or its documentation is prohibited.

2) SMSQ/E will be made available, as source code only, to any person who so requests it. The request must be made to the "registrar" (me). The source code will be sent via CD ROM, thus the request must be accompanied by 3 IRCs, else it will be ignored.

The SMSQ/E that will be so made available is the SMSQ/E as it stands NOW. Any future changes/additions/modifications/adaptions to this code may, or may not, be excluded from the offcial release version, since the authors of such changes/additions/modifications/adaptions may state whether they want their source code to be included in the official distribution, or not.

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost.. **£24**

************NEW*************

superHermes

A major hardware upgrade for the QL

All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software)£90 (£92/£93)
IBM AT UK layout Keyboard..............£11 (£13/£15)
Serial mouse ....£8 (£8.50/£9)
Capslock/scrollock LED ....£1 (£1.50/£1.50)
Keyboard or mouse lead ....£3 (£3.50/£3.50)
High speed serial (ser3) lead ....£4 (£4.50/£4.50)

Hermes available for £25 (£26/£27) **Working ser1/2 and independent input, debounced keyboard.**

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.
Cost (incl keyboard lead) ....£53 (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS
DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.
First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKL...£40 (£41/£43) **MKII...£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL
A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.
Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq...........**£39** (£40/£41)
4mbytes RomDisq.............**£65**(£66/£67)
8 mbytes RomDisq.........**£98** (£99/£100)
Aurora adaptor...................**£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost .... **£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)
2 amp (for 8 relays, small motors) .... £40 (£43/£44)
4 amp total (for motors etc) .... £45 (£48/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver)...£25 (£28/£29)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required........£25 (£27/£28)
Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting................£30 (£31/£32)
Temp probe (-40°C to +125°C)..............£10 (£10.50/£11)
Connector for four temp probes..............£10 (£10.50/£11)
Data sheets..£2 (£2.50/£3)
Control software & manual (for all I/F) ....£2 (£2.50/£3)

QL SPARES

Keyboard membrane .... no longer on sale
1377 PAL .... £3 (£3.50/£4)
Circuit diagrams .... £3 (£3.50/£4)
68008 cpu or 8049 IPC...................................£8 (£8.50/£9)
8301/8302 or JM ROM or serial lead........£10 (£10.50/£11)
Power supply (sea mail overseas)...................£12 (£17/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address./ postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details
10 May 01

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254   Fax/BBS: 01442-828255
tony@firshman.demon.co.uk   http://www.firshman.demon.co.uk

3) No distribution of SMSQ/E may be SOLD, except for the official distribution. This interdiction includes that of including and distributing SMSQ/E in Public domain libraries.

Official distributions will be sold in compiled (binary) form, possibly together with the official distribution as source code. For such sales, for the time being, two distributors/resellers, namely Jochen MERZ (JMS) and Roy WOOD (QBRANCH) have been appointed by the copyright holder. Resellers provide support for the versions sold by them. Binary, i.e. compiled, versions of SMSQ/E may not be distributed other than through the distributors.

4) The registrar, i.e. me, will maintain official distributions of SMSQ/E, in binary and source code form, one for each machine on which SMSQ/E may run.

5) Any person may make any changes/additions/modifications/adaptions to the source code he feels like. Any person may give away to others the modification he thus made, including the official distribution in source code form only, provided this is made ENTIRELY FOR FREE - no charges, not even copying charges, or charges for the media on which this is distributed, may be levied.

This distribution of the source code including the changes/additions/modifications/adaptions made by any author may not be made in electronic form other than on a physical disk.

Distribution of the changes/additions may be in binary (compiled) form, provided that the original and/or official version of SMSQ/E, which is copyright - TTebby, is not distributed in binary form as well.

6) Any changes/additions/modifications/adaptions may be proposed by their author(s) to the registrar for inclusion in the official distributions of SMSQ/E. The registrar is not obliged to inclue any proposed changes/additions/modifications/adaptions in official versions.

When making such a proposal for inclusion, the author of the changes/additions/modifications/adaptions may state whether his contribution:
- is to be distributed in the same way as the official version, or
- is to be made only in the compiled (binary) form of the official distribution, or
- is to be made alongside, but not included in, the official distribution.

Failing such a statement, the inclusion will be contained in the compiled and the source code versions.

By submitting code to the registrar, the author agrees to the limitations as set out herein.

7) Authors retain copyright over their changes/additions/modifications/adaptions, but when and by submitting them to the registrar, they explicitly agree that, if they are accepted in any official distribution (under the provisions hereof), they may be included in all other future distributions (in other words, you can't submit something, which is included, and then some months later attempt to withdraw it).

By and when submitting proposed changes/additions/modifications/adaptions to the registrar, author(s) of such changes/additions/modifications/adaptions also agree that others may, in the future, bring about changes/additions/modifications/adaptions to the code of any changes/additions/modifications/adaptions such as submitted to the registrar.

Copyright of the author(s) of any changes/additions/modifications/adaptions will be acknowledged in the official distribution.

8) For testing purposes only, authors having made one or several changes/additions/modifications/adaptions of SMSQ/E may, as an exception to the prohibition of distributing code stemming from the official release version in binary form as mentioned above, give away binary version of their code, together with binary versions of SMSQ/E, to not more than 10 persons in total (whatever the number of test versions), provided that the persons receiving these test versions agree to destroy them:
- after a period of 2 months, or
- at the time the changes/additions/modifications/adaptions for the test version are included in the official release version, or
- when they are notified by the author that a stable version is now available
whatever comes first.

9) A styleguide will be maintained by the registrar, to which any addition must adhere. The styleguide will be part of the official distribution.

End of official statement.

I have already been approached privately by persons finding this licence unfair, for two reasons which, at the time of writing, hadn't been aired in the QI-users list (or anywhere else, to my knowledge) yet:
1) When a new author adds some code to make SMSQ/E better, only the resellers (and Tony Tebby) see some profit from it.
2) Authors making additions etc are still prohibited from distributing binary versions of SMSQ/E, even for free.
This, some argued, will stifle the development of SMSQ/E since no author will agree to that.
I personally do not think that these arguments are really a hindrance to the future of SMSQ/E - the resellers provide support when selling the binary versions, hence they should get some money, and some restriction in the binary distribution is acceptable for that. We also want to avoid having too many unofficial versions floating around.
If you have any strong feelings in this respect, please let me know!

I have now received the source code from Tony Tebby, and, as soon as this licence is hammered out and I have had a chance to put everything in order, I'll start sending it out. I'm now taking orders...

Wolfgang

Clocking In - Part 3
David Denham

In this part of the series we are going to have a look at a date calculation formula called Zeller's Congruence.

Using this mathematical formula enables us to calculate which day of the week corresponds to a given date – in other words, we can write a program to tell us on which day of the week we were born, what day of the week next year's Christmas day will fall on and so on.

Since the present calendar only came into being in 1582 (adopted for use in Britain and America in 1752) and will accumulate an error of one day by the year 4902, the calendar is only valid for use from 1582 to 4902. This is hardly likely to be a problem!

One unusual aspect of the Zeller's congruence formula is that although we would think of January to December as month numbers 1 to 12 respectively, in fact the formula treats March as month number 1, December as month number 10, January as month number 11 and February as month number 12. Thus

1st January is in fact the beginning of the eleventh month of the previous year. Not a problem as long as we are aware of this.
We need the following information for the formula:

day: day number of the month (1 to 28, 29, 30 or 31)
month: (Jan=11,Feb=12,March=1 etc as described above)
year: the last two digits of the year number (0-99)
cent: century number (1st two digits of year number)

This formula returns a number (dn) from 0 to 6, which corresponds to Sunday (0) to Saturday (6) respectively.

```
dn=( ((2.6*month)-0.2) +
day + year + (year/4) +
(cent/4) - (2*cent) ) MOD 7
```

In practice, the routine can return a negative number so you will see some programs (including examples below) which add a large positive number divisible by 7 before carrying out the "MOD 7" calculation

just to ensure that the number is already positive (adding a number divisible by 7 should make no difference to the MOD 7 action).

The simplest application of Zeller's Congruence is a little program in Listing 1 which asks you to enter a date then tells you what day of the week that date falls on. For example, try entering 25th December 2002 to see what day Christmas day falls on this year. Incidentally, when entering month numbers, just enter 1 for January, 2 for February, 3 for March and so on. The program takes care of adjusting to the other numbering system described above. In addition to telling you the day name, it also tells you if the year in question is a leap year or not, since it needs to work this out as part of the other calculations.

You may find it useful to add a few lines to this program to error trap the entries – make sure that the numbers entered are positive whole numbers, that no invalid numbers of days are entered (e.g. February cannot have more than 29 days) and that the year numbers entered are in the range 1582 to 4902.

Listing 1 - Day Of The Week

```
100 REMark day of the week, using Zeller's Congruence
110 :
120 INPUT'Day (0-31)         > ';day1
130 INPUT'Month (1-12)       > ';month
140 REMark month number should be Jan=11,Feb=12,Mar=1,Dec=10
150 REMark adjust the month number accordingly
160 IF month<3 THEN adj_month=10+month:ELSE adj_month=month-2
170 INPUT'Year (1582-4902) > ';year
180 :
190 REMark decide if a leap year
200 IF ((year MOD 4)=0 AND NOT((year MOD 100)=0)) OR (year MOD 400) = 0 THEN PRINT year;'
    is a leap year': REMark leap year = yes
210 adj_year = year
220 REMark Jan and Feb are months 11 and 12 of previous year
230 IF adj_month > 10 THEN adj_year=adj_year-1
240 century = adj_year DIV 100 : REMark the century number  15 to 49
250 year_in_cent = adj_year MOD 100 : REMark year within century 00 to 99
260 :
270 REMark this is Zeller's Congruence
280 day_no=(INT(2.6*adj_month-.2)+day1+year_in_cent+(year_in_cent DIV 4)+(century DIV
    4)-2*century+700)MOD 7
290 PRINT day1;'/';month;'/';year;' (DD/MM/YY) is a ';
300 SELect ON day_no
310   =0 : PRINT 'Sunday'
320   =1 : PRINT 'Monday'
330   =2 : PRINT 'Tuesday'
340   =3 : PRINT 'Wednesday'
350   =4 : PRINT 'Thursday'
360   =5 : PRINT 'Friday'
370   =6 : PRINT 'Saturday'
380 END SELect
```

The next useful step is to produce a program which calculates dates to the extent that it can print calendars for us. I wrote such a program, but I could not get it to work reliably at the turn of the century, so the editor sent me a similar program he'd written which does seem to work as intended, which I present here in Listing 2 slightly shortened.

As it stands, the program just prints the calendar to the screen. Change the

PRINT statements to

PRINT #channel_no, where

channel_no is the number of a channel opened to the printer, e.g. OPEN#3,SER1 to make paper printouts. If you want to pause the output between months, simply add a PAUSE statement with 585 PAUSE, then you'll need to press a key to print the next month.

Listing 2 - Calendar

```
100 REMark QL Calendar program, uses Zeller's Congruence
110 :
120 WINDOW 448,200,32,16 : CLS
130 :
140 DIM month$(12,10),days_in_month%(12)
150 RESTORE
160 FOR a = 1 TO 12 : READ month$(a)
170 FOR a = 1 TO 12 : READ days_in_month%(a)
180 :
190 REPeat get_year
200    INPUT'Enter year (e.g. 2002):';year%
210    IF year% >= 1752 AND year% <=4902 THEN EXIT get_year
220 END REPeat get_year
```

```
230 CLS
240 :
250 REMark find day of week of January 1st
260 LET century  = (year%-1) DIV 100 : REMark century
270 LET year1    = (year%-1) MOD 100 : REMark last 2 digits of year
280 day_no       = 799+year1+(year1 DIV 4)+(century DIV 4)-(2*century)
290 day_no       = -(day_no MOD 7)
300 :
310 leap_year    = 0 : REMark is this a leap year?
320 IF ((year% MOD 4)=0 AND NOT((year% MOD 100)=0)) OR (year% MOD 400)=0 THEN
    leap_year = 1
330 :
340 FOR month_no = 1 TO 12
350   PRINT \\ TO 11;month$(month_no);' ';year%
360   PRINT \ TO 4;'SUN MON TUE WED THU FRI SAT'
370   IF month_no = 2 THEN
380     REMark February has 29 days in leap years
390     days_in_month%(month_no) = days_in_month%(month_no)+leap_year
400   END IF
410   FOR line_no = 1 TO 6
420     FOR column_no = 1 TO 7
430       day_no = day_no + 1
440       IF day_no > days_in_month%(month_no) THEN
450         start_next = column_no : EXIT line_no : REMark end of month
460       END IF
470       IF day_no > 0 THEN
480         PRINT TO column_no*4;
490         IF day_no < 10 THEN PRINT ' ';
500         PRINT day_no;
510       END IF
520     END FOR column_no
530     start_next = 1 : REMark previous ended on Saturday
540     PRINT : REMark go to start of next line
550     IF day_no = days_in_month%(month_no) THEN EXIT line_no
560   END FOR line_no
570   day_no = 1 - start_next : REMark where does next month start
580   PRINT
590 END FOR month_no
600 :
610 REMark data for names of months, and days per month
620 DATA 'JANUARY','FEBRUARY','MARCH','APRIL','MAY','JUNE'
630 DATA 'JULY','AUGUST','SEPTEMBER','OCTOBER','NOVEMBER','DECEMBER'
640 DATA 31,28,31,30,31,30,31,31,30,31,30,31
```

If you want to adapt the program to only print certain months of the year, you'll need to ask the user to enter the number of the month to print from, and the number of the month to print to. Within the FOR month_no loop, all PRINT statements will need to be modified with an IF clause like:

```
IF month_no >=
print_from_month AND
month_no <=
print_from_month THEN
PRINT...
```

The printout is a fairly basic monthly format calendar, as shown in figure 1. Have fun changing it to different formats!

Figure 1 - sample output from calendar program

```
================================
```

```
            MAY 2002
SUN MON TUE WED THU FRI SAT
                 1   2   3   4
  5   6   7   8   9  10  11
 12  13  14  15  16  17  18
 19  20  21  22  23  24  25
 26  27  28  29  30  31
```
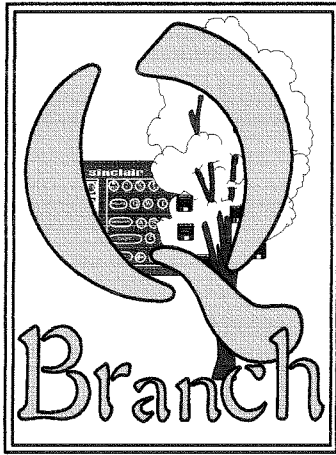
```
            JUNE 2002
SUN MON TUE WED THU FRI SAT
                             1
  2   3   4   5   6   7   8
  9  10  11  12  13  14  15
 16  17  18  19  20  21  22
 23  24  25  26  27  28  29
 30
```

```
================================
```

Finally, Listing 3 is a short program to help calculate the date of Easter Sunday for a given year. This one does not use Zeller's congruence and is included as a good example of

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

New Fax Number ▲

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor
Demo version £ 1.50 + postage
Full Registered version £ 18.00

¶ Version 2.03 available now !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II	£ 100.00
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 45.00 *
Recycled Aurora	See over*
Qubide	See over
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

Qubide upgrades to version 2.01 £ 8.00

SMSQ/E   Now Only £ 32.00

Gold Card / Atari / QXL Version

Various Atari versions : call for details

QPC 2 v3 full colour version!

Now Only £ 65.00

Upgrades from QPC 1 £ 46.00 (return master disk)
Upgrades from QPC 2 v1 £ 30.00
Upgrades from QPC 2 v2 £ 14.00
If Bought/upgraded after 15/10/01 £ 5.00

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E
or QPC 2 upgrade !

Q Branch Programs

The Knight Safe 3	£ 35.00	Q - Route v1.08C	£ 25.00
upgrades from previous versions	£ 5.00	Route finding programme	
Q - Count	£ 25.00	The Fractal Collection	£ 35.00
Pointer driven home accounting			

Post and Packing is included with Software Prices

how easy certain calculations are once you know the base formula.

Easter Sunday occurs on the first Sunday following the first full moon which occurs on or after 21st March. The actual formula for calculating the date on which Easter Sunday falls is taken from one of the classic programmers' bibles, Donald Knuth's 'Fundamental Algorithms" published by Addison Wesley Publishing, ISBN 0-201-03821-8, in case you'd like to try to obtain a copy! In fact, the algorithm is from the 16th century from the work of Aloysius Lilius and Christopher Clavius.

First we calculate the number of the year in the 19 year Metonic cycle (a period of 235 lunar months or about 19 years, at the end of which the phases of the moon recur in the same order and in the same days as in the preceding cycle) in line 150. Then we calculate the century number – 1999 would be 20th century but 2002 is 21st century – who can remember or work out which century was 2000?

Line 170 makes a little correction to allow for the fact that at the turn of each century, only one in four is a leap year (1700, 1800 and 1900 were not, 2000 was a leap year). Line 180 is designed to synchronise Easter with the moon's orbit. Line 190 locates the sunday.

Line 200 to 230 are concerned with the epact, the excess of time of about 11 days of the solar year compared to the calendar year.

Next we set about finding the full moon (remember that the Easter Sunday is the Sunday following the first full moon on or after the 21st March. Line 260 winds us on to the Sunday itself, but since the value of find_full_moon will roll on into April if Easter Sunday rolls past the end of March, we need to subtract 31st March from this value if in April – the routine might try to tell us that 15th April was in fact 46th March otherwise!

If you'd like to generate a list of Easter Sunday dates rather than calculating each individually, add a FOR loop which gives a range of values for the variable "year" instead of the INPUT statement shown, e.g.

FOR y=2000 to 2099 and

LET year = y

Listing 3 - Easter Sunday calculator

```
100 REMark calculate Easter day date
110 REMark from Knuth's "Fundamental Algorithms"
120 :
130 CLS
140 INPUT'Year for which Easter is required ';year
150 golden = (year MOD 19) + 1
160 century = INT(year/100)+1
170 leap_correction = INT(3*century/4) – 12
180 moon_orbit = INT((8*century+5)/25) – 5
190 sunday = INT(5*year/4) – leap_correction – 10
200 epact = 11*golden + 20 + moon_orbit – leap_correction
210 IF epact < 0 THEN epact = epact + 30
220 epact = epact MOD 30
230 IF (epact = 24) OR ((epact = 25) AND (golden > 11)) THEN epact = epact + 1
240 find_full_moon = 44 – epact
250 IF (find_full_moon < 21) THEN find_full_moon = find_full_moon + 30
260 find_full_moon = find_full_moon + 7 – ((sunday+find_full_moon)MOD 7)
270 CLS : PRINT'Easter day is ';
280 IF find_full_moon <= 31 THEN
290    PRINT'March ';find_full_moon
300 ELSE
310    PRINT'April ';find_full_moon – 31
320 END IF
```

That brings this series to a close unless someone has more material or any correc- tions or improvements to con- tribute. I hope you have found it useful, I have enjoyed writing it and all that I have learned while preparing it.

A short visit of XMenu - Part 1

Jérôme Grimbert

The Xmenu library comes with a config block which allows the end user customise the appearance of the PE items managed by the Xmenu library. The customised colour is accessed by calling the DefaultColourSet() function. If a programmer does not want to use the config block or want to have their handling, he only has to provide a pointer to a colourSet structure (this structure is defined in xmenudef.h file), and of course to fill it as he wants. All the displaying functions take this pointer as first parameter, and are able to work in mode 8 too (if the primary is big enough to accomodate the bigger text, otherwise the functions just return simulating an ESC [for the function which gives back a value]).

Message Report
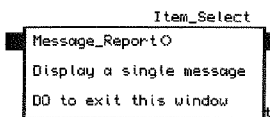
Message_Report() is very simple: the additional parameter is just a string to be displayed until Enter/Do is performed on the popup (ESC also works). Message_Report opens a secondary window, so the primary must already exist.

For the case when there is no primary, Startup_Report() is the one to call instead of Message_Report().

One interesting feature of Xmenu is that the string can contain chr$(10) (LF) to indicate a multi-line message.

String Edit
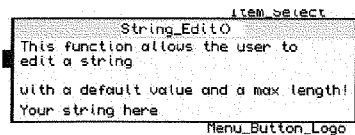
String_Edit() allows to edit a string. The additional parameters are:
- a title for the window
- an explanation text, which can be multi-lines
- a default value
- a max length for the string
- the current value

Upon exit, an integer is returned which is the length of the actual string, unless ESC was used in which case -1 is returned.

XDialog
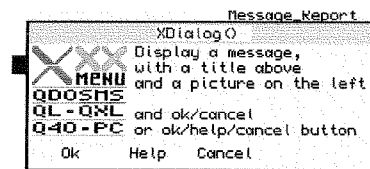
XDialog() was inspired by the X11 dialog where there is usually an icon on the left of some text and some classical choice is offered. For our favorite system, the icon has been replaced by a sprite. The additional parameters are:
- a title for the window
- a message
- a sprite
- a boolean (0/!0) to choose between ok/cancel or ok/cancel/help as items

Upon exit, an integer is returned:
-1 if ESC was used to exit
0 if Ok was chosen
1 if Cancel was chosen
2 if Help was chosen

Button
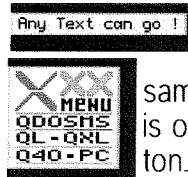
Because putting an application to sleep as a button is always the same, but can be different when there is or is not a button frame, Menu_Button_Text() and Menu_Button_Logo() are here to avoid reinventing the wheel each time. The additional parameters are:
- the main window definition (the one to hide until the button is removed and the application is restored)
- either a text or a sprite for the button (according to the called function)

If there is a button frame, the button will get inserted into it, otherwise the 'classical' floating button will be available.

Item Select
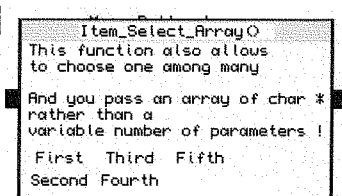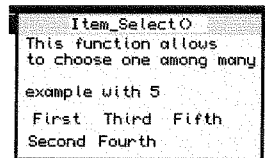
Item_Select() and Item_Select_Array() allow the choice of option among many. The additional parameters are:
- a title for the window
- an explanation message which can be multi-line
- the number of options

Then Item_Select() has a string for each possible option (using a variable number of arguments!) whereas Item_Select_Array() has a single pointer to an array of string (also one for each option).

As usual, if ESC is used to exit, -1 is returned, otherwise the number of the choosen option is returned (0 being for the first option).

The programmer does not have to worry about the size of the window and has no control on the relative positions of all items: all is done by the Xmenu library (it make things simpler for quick code, but takes away some freedom of design with the burden!).

List_Select
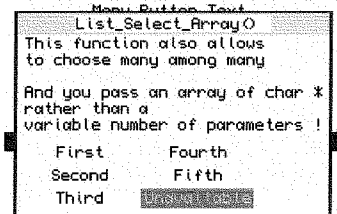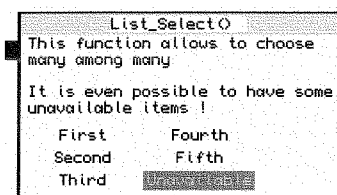
List_Select() and List_Select_Array() allow to select many among many options. There is one additional parameter from Item_Select: a status list which indicates the initial (and after the function call, the final) status of each option. The PE values are directly used in this list. The returned value is the number of selected items.

Radio Select
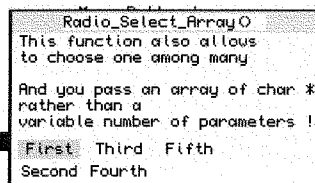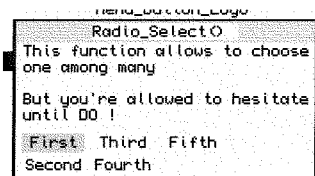
Radio_Select() and Radio_Select_Array() allow to choose one among many options, but with a radio-button behavior. There is one additional parameter from Item_Select: the initial item which is selected. The returned value can be -1 in case of ESC, or the number of the radio button which ends up as being selected.

Radio_Select_Range() saves the programmer from writing a full list of incrementing values, instead only the minimal and maximal values are provided with the step to use between the value. It works only with integer values.

| Radio_Range_Select()								
512	4608	8704	12800	16896	20992	25088	29184	33280
1024	5120	9216	13312	17408	21504	25600	29696	33792
1536	5632	9728	13824	17920	22016	26112	30208	34304
2048	6144	10240	14336	18432	22528	26624	30720	34816
2560	6656	10752	14848	18944	23040	27136	31232	35328
3072	7168	11264	15360	19456	23552	27648	31744	35840
3584	7680	11776	15872	19968	24064	28160	32256	36352
4096	8192	12288	16384	20480	24576	28672	32768	36864

Non displayable functions

Adapting window definition

- LimitGeometry() allows limiting of the maximal size of a primary, thus allowing a programmer to make an universal huge setting in the default value of the definition and adjusting to the actual size of the display (thus allowing 1280x960 windows on QPC and restricting to 1024x512 on Q40, while stopping at 512x256 on normal QL). Best used with DisplayXSize() and Display YSize().
- SetInfoColour() applies the choices of colour from a colour_set over an information 'window' definition (it's a sub-part of a window), allowing the user to really customise the colour in the application. SetWindowColour() performs the same for the (primary) window definition.

Adapting sprites

SmartHighColour() is a function which adapts a high-colour (16 bits) sprite to either the Q40 mode or the QPC mode. Because high-colour sprites are heavy for memory, one might not want to double the memory consumption to have both the Q40 and QPC version of the same sprite. So this function just performs the adaption of one mode to the other, when portability, low memory and high-colour are wanted at the same time.

Getting screen information

- DisplayXSize() and DisplayYSize() returns the actual size of the screen, until a primary is drawn.
- Get_NativeForm() return the form that a sprite must use to be considerated as Native (works for mode 4, mode 8, Q40 and QPC; Aurora is unknown due to lack of information).
- Get_CharSize() gives the length in pixels of each character, if you ever need to know it. (6 unless in mode 8 where it is 12 !).
- Check_Size() tells you without crashing if a window can be drawn or not.

Getting sprite information

Get_GraphincInfo extracts the size and origin information from a sprite, taking into account the actual mode of the display and the various modes in the linked chain of a sprite (because the first sprite definition might not be the one that will be used by the system!).

Logo

The Xmenu Library comes with a sprite (64x60), it's just xmenu_logo. You can see it on the XDialog picture.

Why this visit?

Now that you know what's provided by the Xmenu library, it might be easier to perform some mundane tasks and concentrate on the important part: your application! So let's go back to the explanation of PE in C in the next issue.

Letter-Box

Simon N Goodwin writes:
The article 'Which machine am I running on' last issue was a bit frustrating as it makes several assumptions that clash with Qdos Classic and Amiga Qdos conventions that have been established and freely available for years. The variable at offset 161 in Mark's systems (QDOS Classic etc -DJ) matches the document, but the one at 167 does not, and this is the first time I've even seen it mentioned. Contrary to the article, it doesn't work on my Gold Card – maybe it is SMS only?

It also seems a pity that the article didn't mention or take account of QLAY or the various Thors, QEm or UQLX, etc. While I understand that Marcel may not be interested in those, they ought not to be excluded from QL Today and so I think it was a mistake not to mention or allow for them. Rather than present that information as if it was complete, I'd prefer to see suggestions about how it can be extended without running the risk of people using the same code for more than one system.

PROCESSOR_ASM (in the free Qdos Classic sources) can work out the value for 161 in a general way for all 68Ks. I can see no general way to add or abstract the information attributed to byte 167, but there are ad-hoc ways that work on most emulators and add-ons so the list could be extended to include the whole Qdos community, not just the SMS subset. It is not sensible for QL Today to close out any potential readers; the article should at least have mentioned the obvious gaps.

Reply from the Editor:
In volume 6 issue 5 an article from Marcel Kilgus and I described the use of facilities in SMSQ/E based systems which tell you what kind of processor, display and so on is available on which type of machine. Simon Goodwin, John Perry and others pointed out that this article contained no information relevant to QDOS users (e.g. Amiga QDOS users).

As far as I was aware, there was no support in any version of QDOS for these facilities and as I did not know how SMSQ/E obtained this information from the hardware on the machine it was running on, I did not even know if these facilities could be implemented in QDOS.

I circulated an email to the authors of the various QL emulators and Minerva asking if their products had any support for anything like this or if they might at some point in the future do so. I got just one reply – asking for a copy of the article as he hadn't seen it. So do I take it that the answer is a firm NO?

So I am now going to issue a public appeal. Jochen does not have enough knowledge of QDOS on other systems (he's been an SMSQ/E man for years of course) and I don't really have enough knowledge of either at this level (which is why I got Marcel to co-operate on the original article) – does anyone reading this have the necessary knowledge to help us with this one? Is there any software package already out there which will provide some or all of these facilities? Simon is correct to point out we should not exclude QDOS users totally. I'm told the Gold Card and SuperGold Card patch QDOS to some extent to provide processor information at least, so it is probably possible in QDOS with the right knowledge. And with the SMSQ/E development becoming a bit more open, with access to the source codes now possible as the SMSQ/E Registrar (Wolfgang Lenerz) has described, someone with knowledge of the operating system at this level may yet be able to bring such facilities to 'vanilla' QDOS. Please.

Last Minute News

PQIV QXL and QPC support

QL Today reported in the first News Section about a new version of PQIV from Claus Graf. Unfortunately, it still does not support QXL and QPC screen modes.

Marcel Kilgus has find an easy way to patch the new version of PQIV so that all QLers with hi-colour graphics mode can benefit from the program.

The patch can be downloaded from the public archive:

http://www.mail-archive.com/ql-users%40nvg.ntnu.no/ msg05967.html

QPC2 News

The next sub-version of QPC2 Version 3 will hopefully be released by the time you read this News (i.e. first chance at the North American QL Show). Marcel is working hard to get it ready for the show.

The new version QPC2 V3.03 will most likely contain the new Window Manager (features described in the previous issue of QL Today) and improved control commands for the DOS device (e.g. DOS_DRIVE$). Marcel also hopes to have a new memory scheme implemented which works like the one implemented on the ATARI TT. This will speed up file access time considerable with large amounts of RAM!

Updates from previous version 3 will be free, for other upgrade prices check the JMS advert.

BYTS OF WOOD

SAW POINTS OFFCUTS AND SNIPPETS
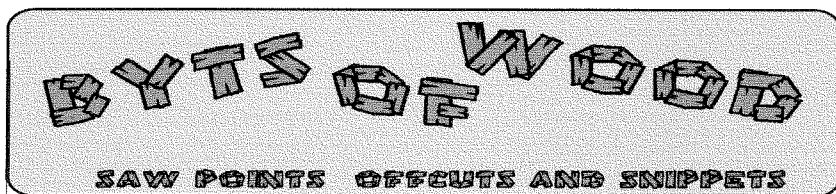
The factory which makes the dots for 'i's and the crosses for 't's has been going into production overdrive over the last two months as Wolfgang Lenerz struggles heroically with the SMSQ/E licence agreement. One thing which has come out of the whole affair is how much people can fail to agree, even when the argument is largely to everyone's benefit. Somehow or other people seem to want to say 'Oh you are going to let us have it a lot cheaper - well that is not good enough give it to me for nothing'.

I have made a lot of comments in the past about free software and I suppose that you all know these very well by now but it seems obvious to me that you can have SMSQ/E for free if that is what you want but there will not be anyone around to support it or provide you with a manual (which some of you will not read anyway) or help you when you have a problem which you cannot either understand or get around. Many of the people who are clamouring for it to be a free system are those for whom reading a mass of source code and understanding it would be a simple task but these are not the average user. I have often remarked that the 'here are the extensions/code - write your own front end' brigade react contemptuously when faced by users who cannot do that. I equally cannot see why some people shun writing some simple programs to perform useful tasks because 'the user could write that themselves in BASIC'. Given that I am one of those who advertise their phone and internet details in the QL press I have a lot of contact with the end user and I can tell you that many of these are just that - users. They use a QL or a QL emulator because they know it and they have no real need to have to learn another system. They cannot write anything and

even a BOOT file is regarded as 'beyond my abilities'. This is not criticism or one-upmanship, just pure observation. In many cases, if they sat down and tried they would get the hang of it very quickly and I do not think that any of them are stupid in the least.

People like Jochen, Tony Firshman and I are here to provide support for the users and we do try to do this with as much patience and goodwill as possible. I have been known to get upset when people ask the same question over and over - especially when the answer is in the manual and they have not bothered to read it - but it is rare. This support work is completely unpaid and, to return to the SMSQ/E question, the small fee that is left over after we have paid Tony Tebby, printed the manuals and provided the disk goes a little way towards paying us for all the work.

It is Not a Horse It Is Just Two Coconuts.

The whole argument between some of the people who want a completely free SMSQ/E and the 'gang of five', as Wolfgang

Lenerz calls the five people who were present at the Eindhoven meeting which thrashed out the start of the licence for SMSQ/E has begun to turn into one of those Monty Python style arguments. People seem to find ever more esoteric examples and exceptions and I begin to wonder if they are arguing in their spare time.

Some people argue about one thing and some about another. It has even got to the stage where some people are arguing about the arguments other people have put forward. Maybe two swallows got together and carried all these arguments here because one swallow could not have managed it.

I Liked The System so Much I Bought The Copyright!

One person wanted to pay Tony Tebby 300 Euros to be allowed to give it away for free. Well that is very altruistic but as soon as he has given away more than 30 copies Tony is also making a loss. Some of this is tied up in the particular kind of politics that has emerged in the QL World over the last year or so with one side trying to rubbish the other. Conventional capitalist wisdom is that competition is always a good thing for the consumer and, ipso facto, this kind of competitiveness should make the two sides strive to improve their systems.

Well, sometimes it is and sometimes it isn't. In the wide computer world Sun Microsystems developed Java and the language was eagerly taken up and used on websites. This kind of thing is an ideal situation but some operating systems have developed flavours of Java which differ radically from the original and some browsers

have actively worked against rivals versions.

The suspicion exists (and it it not too unfeasible given the previous behaviour of this party) that a similar ploy may be lurking at the heart of this offer and that is something that we would all do well to reject. The copyright for SMSQ/E is not up for sale here and the whole post of Registrar is more to do with maintaining cross system compatibility than protecting the small revenue that will come from the sale of copies of the system.

All in all the overriding feeling I have got from all of this is a great admiration for Wolfgangs patience, perseverance and tact. Hats off to him I say for not throwing the whole thing back into the ring and going off for a long holiday somewhere that you don't have to wear an anorak.

The Heart of the Matter

I apologise for taking up so much space in this column with this subject but it is one which is of great importance to all who use the QL or its offshoots. Whatever system your code runs on you need to be sure that as much software as possible is compatible with it. If too many applications start to fall over because of new developments then users will, eventually, have to abandon the system in favour of one which is of more use or revert to the previous stable version.

The Colour Drivers are a case in point here. Yes, they can enhance your system, and you can use them from BASIC or by calls directly to the hardware but for the moment the Window Manager has no access to the available colours. That may be about to change but the real point here is that the re-

arrangement of the screen which was needed to get the colours working has meant that many older programs have a corrupted display and many are unusable. When we all started off using the QL we had so little memory that we had to reboot each time we wanted to use a different program but that is not the way it is today and it neither should it be.

It is, however, exactly what users who rely on Text 87, for instance, have to do if they want to run most of their stuff in high colour mode and still use the word processor. Again this situation may change because Fred Toussi, the program's author, has offered to make some changes to the program if a change is made to the SMSQ/E high colour display driver. All we need now is someone who is willing and able to make the changes. No small task I expect.

You can go in and reprogram SMSQ/E, if you have the ability, and change any part of the system to make it run better in some areas than in others but there will always be inconsistencies and conflicts and we really need the utmost co-operation amongst the programmers to make sure that shaving a few nanoseconds of a process or improving a benchmark is not the sole goal of such reprogramming. It may look good in the advert but what is the good of a fast car if it crashes whenever a passenger gets in?

New Bits for Old

In spite of all of the foregoing there is some cause for optimism. I am expecting to get a test version of a new keyboard interface, The Qeyboard, (what else?) soon although I have not yet either seen the item or know the prospective price.

Q-CELT COMPUTING

THE LARGEST SUPPLIER OF PURPOSE MADE QDOS CD-ROMS

The Falconry, Glenmacnass, Glendalough, Co.Wicklow, Republic Of Ireland.
Tel:(+353)404-45319 Fax:(+353)404-45558 Mobile:(+353)86-8100090
Email: qceltcomputing@hotmail.com

THE DILWYN JONES QL COLLECTION - All of the commercial and freeware software ever written by Dilwyn Jones, suffice to say this is excellent value and there are several programs here that no QL user should be without. REDUCED TO £15 !!

THE QL EMULATORS CD - This is a collection of all known QL emulators for various formats, including the PC, Apple Mac, Unix/Linux, Atari and Amiga. It even includes a special version of Tony Tebbys Toolkit II and approximately 1,000 (Yes, one thousand!!) freeware and Public Domain QL programs. SPECIAL : JUST £5!!

THE RELIGION CD-ROM - A huge collection of various texts, books and documentation with religious signifigance. All the major religions of the World are covered - origins, beliefs, etc. as well as sermons and parables. Priced at Just £10

LINEDESIGN CLIPART CD - A full CD of the finest clipart ever produced for the QL. Over 600 Megabytes of Data, and literally hundreds of files, crammed onto just one CD - perfect for designing that flyer or newsletter. Now just £10.

INTERACTIVE FANTASY CD - The work of Phoebus Dokos, this is a great CD for the Adventure Gamer, and contains hundreds of INFOCOM adventure games,.. Lifetime updates are available FREE via email from Phoebus - ask for details. SPECIAL PRICE :- £10 (a percentage of all profits will be donated to charity)

FAMOUS FACES CD –caricatures of the rich & famous in lineDesign format. £10.

THE ZEXCEL SPECTRUM EMULATOR CD - A CD based on the popular ZX Spectrum Computer emulator - play all your old Spectrum favourites on your QL - thousands of Games are included, Utility Programs, Wp's, Desktop Publishing etc. The latest version of ZeXcel for the QL is also included - months of fun for only £10!!

WORLD OF Z88 CD – for the Cambridge Z88 Computer - almost 2,000 files, including MS-DOS and Windows Z88 Emulators, EPROM Images, Games, Utilities, the Z88 User Library, Sourcebook, etc. Just £8!!

QL DOCUMENTATION CD – A collection of doc's for the QL on all available hardware and some hard to find information. Includes all miracle manuals! £10

Above Prices do not include P&P – please add £1 per disk. We accept Cheques in Sterling made payable to "D.Branagh" and Major credit cards by prior arrangement – please email or phone for details. We also stock rare pieces of QL hardware & Software – Call us!
Please allow up to 14 days for delivery. E&OE.

Dave Park does keep threatening to put a picture up on his site so maybe that will be there by the time you read this.

The new Qubide is also making progress and has, I gather, reached the pre-production stage so the hardware scene may well be set for a makeover. (Changing Roms - for my English readers) Rumour has it the that new IDE device will have a built-in Ethernet connection so I should be able to connect the Aurora machine to the PC and that is an interesting prospect.

Of course there is always the old bugbear of having to have proper software support although I had heard that Jonathan Dent's TCP/IP stack was already up to the task. (Again this is only hearsay and conjecture. I am sure that someone with more knowledge than me will put it right). I hope to have more first hand news of all of this after I return from the US show in June because Nasta - who has a hand in these projects - will be on hand to enlighten me.

A QL based Ethernet for the Q40 would be a useful item. I know you can do it from LINUX but, as far as I know there is not QDOS/SMSQ equivalent. In the general run of things the Aurora machine is still hooked up but is very much a poor cousin to QPC2 and the Q40 and is very underused.

QPC2 tends to do most of my QL work with the Q40 running the magazine subscriptions and Q Branch Database. Before I start getting any (more) Q40 hate mail this is purely down to the fact that the PC is usually already on and running all of the internet tasks, moving my live music tapes onto CD etc. The other advantage of QPC2, of course, is the ability to run in QL Colour mode and run Text

87 which the Q40 cannot do without being limited to 512 x 256. But I have mentioned this before - I should not go getting too recursive here.

Duncan Says
And talking of systems and the Q40.....
At the recent Manchester AGM I chatting to the, very knowledgable, Duncan Neithercut and I were chatting about problems with compatibility. He said that he had just bought a Q60. We talked about the Pro-WesS and QLIB problems and I said that I had lost a lot of heart in the Q40 because these problems were never really addressed - just denied. I got an email from Duncan which I reproduce here in full.

"Hello, I think you have mentioned to me you have experienced the same difficulty as I have in getting ProWesS to run reliably on the Q40 with the patched version of SMSQ/E 2.98 or 2.99 and especially when QLib is run.

I may have a solution that I have come across while transferring things over to the Q60 I have recently acquired. The Q60 support disks came with CacheModes_rext which is from Mark Swift and is on his web site. It gives more control over the caches and gives slightly different results from the SMSQ/E cache command I have used previously. This needs to be LRESPRed at the start of the boot file and the WRITETHROUGH command used at the start of the boot file I also use the SMSQ/E cache_off command as well. At the end of the boot Cache_on and either COPYBACK or WRITETHROUGH is used again to get some speed. If COPYBACK is enabled QLIB

and some QLiberated programs still run into trouble but with WRITETHROUGH the QIiberated programs I have seem OK. QLib_obj itself does not reliably run until the configuration values for stak and heap are X4 increased beyond their default values.

Once that is done and WRITE-THROUGH enabled I can compile SBASIC up to 80k and print from line design on the Q40 or Q60.
If the above is coherent it may be of use
Duncan"

I have made a brief attempt at testing the concepts involved here but it is not as thoroughly tested as I would like. I will try to report on this more in the next issue. One thing that does occur to me is the fact that having to load and manipulate cache software is not a satisfactory solution - especially given the foregoing comments on user abilities. It is absolutely imperative that the system is of use to someone who does not want to concern himself with these things too deeply.

We can dismiss these people if we like and say that the system is not for the faint hearted and those whose programming skills are not up to the task of keeping the system functioning - and then we will have so few users that it will not be worthwhile to continue. That way Linux lies.

Q40isms again
It would also seem that the version of the Q40 being sold by D&D systems is not the same as the one that Q Branch was selling. Apparently this is the Q40i and the Q40, as such, is 'no longer supported'. Nice of them to tell us, eh? I have not had any details of the difference between the two ver-

sions (except that the Q40i supports 128Mb RAM) but Derek hinted that I should get a Q60 and the problems I had would go away. I think we need some more details here if the user is not to feel he is being kept deliberately in the dark.

If it was felt necessary to change more than just the RAM capacity on the Q40 then what was changed, why and, more to the point I would say, is there any way that those of us who have a Q40 can upgrade it to a Q40i without having to buy a new board? Time to 'fress up' guys.

Headhunters

There seem to be a lot of collectors around these days. A few years ago it was hard to give a QL away and I know for a fact that several of the old QL User Guides went straight to the tip after a QL workshop when no-one wanted to buy or even have them from free from a Bring and Buy sale. I have about 10 standard QLs in working order here and I have passed some of them on to users who have either stripped theirs down in the past to build a new, cased, system or got rid of them when they moved onto a PC (with or without a QXL/QPC emulator).

Collectors tempt you into offe-

ring the QLs for silly prices but, on the whole, they just want to put the things into glass cases and not get them up and running. One person who contacted me wanted some microdrive software so he could run it up but admitted that, after he had had it going, he would put it into the display case and that would be it.

Someone did tell me once how many QLs were made (I instantly forgot it again) and it was a pretty impressive number. Now there are not really so many of the original boxes around. When I stripped two down and built them into cases I just threw the odd bits away and I suspect there were many others who did the same thing. For a while there was not a standard QL to be seen at Q Branch Towers. I was very pleased when Jamie Muggleton, who runs the office for Judie Tzukie (a singer and songwriter of some excellence) told me that he had a complete and hardly used QL in an original box in the office. This machine is now the one that I use to make the cover disks on. I have accumulated a few more since.

Many of us are collectors too of a kind. Over the years of running Q Branch I have managed to accumulate all of the issues of QL World, QL User IQLR etc and have them in the original binders too. I know that Urs König has managed to collect an impressive array of QLs of various builds. However I think that we are all reading this and interested in the QLs history and future because we like what it can do and we like how it does it and that is entirely how it should be.

Across the Pond

As I finish this I am preparing for the next US show. In fact the first people to read these words may be the attendees of the Washington DC show. There should be some interest there for Jim Hunkins QDT programs and Nasta's projects. I hope to be able to give some insight into this and other things in the next issue.

ERRATA

These two screen dumps were omitted from George Gwilt's TurboPTR and QMenu article in QL Today Volume 6 Issue 5.

The QL Show Agenda

QL Meeting - (NL) Eindhoven

Saturday, 22nd of June, 10:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

The usual range of dealers will be there, e.g QBranch, TF Services and J-M-S

Quanta workshop - (GB) Byfleet

Sunday, 22nd of September
Same venue as every year:
The Byfleet Village Hall.

German QL Show - (D) Berchtesgaden

Two-Day Event, Sat./Sun, 5th/6th October
Same venue as last year: Hotel Schwabenwirt
Koenigsseer Str. 1, D-83471 Berchtesgaden
(phone +49-8652-2011, fax -1706, no email)

How to get there: Only 150m from the big railway-/bus-station in Berchtesgaden: You cross the street and the bridge to Koenigssee (not Schoenau, it leads to the graves!) and then see at the left side the yellow hotel with some green trees around in a nice beer-garden and a good cinema inside. Very easy to find! Parking near the hotel.

Further information via **F.Oertel.BGD@t-online.de** or +49-8652-94871-2 or -3   or +49-170 40 65 272 mobile
The meeting starts on friday afternoon, 6.10.2001 and ends on sunday, 8.10.2001 in the morning (departure, sightseeing). The main-event is on Saturday, 7.10.2002. QL-enthusiasts, I hope, will come from around Germany (especially Munich), Great Britain, Austria (esp. Vienna, Salzburg), Netherlands and even USA.
For those who like to stay in this wonderful region, here are some hints, for getting rooms too:
www.berchtesgadener-land.com or **www.berchtesgaden-last-minute.de** or **www.salzburg.com**

We have a lot of proposals for nice trips and wanderings - also for the whole family:
In Berchtesgaden there is a fantastic adventure-swimmingpool, two cinemas, a famous saltmine, castles, fortresses, the national park, the historic Obersalzberg, in former times Hitlers mountain (see the bunkers and a museum), busses and elevators to high mountains, lifting you from 550 m to over 2500m high. Please remember, that you should have good boots and clothing for such excursions into the higher mountains. In harvest there is often brilliant sight far into the alps!!! Take a trip with the ship across the famous kings lake (Koenigssee) with clear water, 200m deep or go to a trip with a mountainbike. See the oldest pipeline, made from wood, for the salty water or go to the salt-museum in the near Bad Reichenhall. It's great! Make your trips by bus or car.
Nearby there is Salzburg, a fine historic town with a big fortress, shopping-streets, museums, a zoo, ...

Driving directions in the next issue.